VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

LABORATORNÍ SCÉNÁŘE POPISUJÍCÍ SYSTÉM DNS
LABORATORY SCENARIOS DESCRIBING DNS SYSTEM

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — **Bc. Peter Sakala**
AUTHOR

VEDOUCÍ PRÁCE — **doc. Ing. Jan Jeřábek, Ph.D.**
SUPERVISOR

BRNO 2018

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Diplomová práce

magisterský navazující studijní obor **Telekomunikační a informační technika**
Ústav telekomunikací

Student: Bc. Peter Sakala ***ID:*** 164388
Ročník: 2 ***Akademický rok:*** 2017/18

NÁZEV TÉMATU:

Laboratorní scénáře popisující systém DNS

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s fungováním systému DNS (Domain Name System) a jeho možnostmi užití v Internetu včetně možných implementací jednotlivých komponent systému. Navrhněte částečně či plně virtualizovanou topologii DNS hierarchie zahrnující DNS koncový, rekurzivní, národní i kořenový server s možnostmi konfigurace jednotlivých komponent a propojení mezi nimi. Navrhněte dva kompletní laboratorní scénáře nad vámi navrženou topologií. Vytvořte a popište vhodné výchozí prostředí a nastavení u každého scénáře. Důraz musí být kladen na vizuální stránku (analýza ve wiresharku, grafické zpracování), vypovídající výstupy a porovnání více možných řešení či přístupů. Vytvořte detailní návody pro oba dva scénáře včetně vzorového řešení.

DOPORUČENÁ LITERATURA:

[1] JEŘÁBEK, Jan. Pokročilé komunikační techniky. verze 2017. Brno: Vysoké učení technické v Brně, 2015. ISBN 978-80-214-4713-4.

[2] Knot DNS (software) [online]. CZ.NIC: 2017. Poslední změna 8.9.2017 [cit. 8.9.2017]. Dostupné z: https://www.knot-dns.cz/

Termín zadání: 5.2.2018 ***Termín odevzdání:*** 21.5.2018

Vedoucí práce: doc. Ing. Jan Jeřábek, Ph.D.
Konzultant:

prof. Ing. Jiří Mišurec, CSc.
předseda oborové rady

ABSTRAKT

Diplomová práca sa zaoberá predovšetkým systémom doménových mien (DNS) a jeho praktickému využitiu. Popisuje hierarchiu doménových mien, typy záznamov, používaný protokol ako aj rozšírenie DNSSEC. Taktiež sú popísané najčastejšie používané implementácie autoritatívnych a rekurzívnych DNS serverov. Bola popísaná virtualizácia, kontajnery a iné nástroje využiteľné v laboratórnych úlohách. V práci boli navrhnuté a vypracované dva laboratórne scenáre vo virtualizovanom prostredí s návodmi pre študentov.

KĽÚČOVÉ SLOVÁ

BIND, dig, DNS, DNSSEC, Docker, doména, Knot DNS, laboratórna úloha, menný server, netem, NSD, Ubuntu, Unbound, VirtualBox, Webmin, Wireshark, zóna

ABSTRACT

The master's thesis deals with Domain Name System (DNS) and its practical use. It describes hierarchy of domain names, resource record types, protocol used, as well as DNSSEC extension. The most utilized implementations of authoritative and recursive DNS servers are presented. Virtualization, containers and other tools with potential use in labs are described. Two lab scenarios in virtualized environment with instructions for students were designed and developed in this thesis.

KEYWORDS

BIND, dig, DNS, DNSSEC, Docker, domain, Knot DNS, lab, name server, netem, NSD, Ubuntu, Unbound, VirtualBox, Webmin, Wireshark, zone

SAKALA, Peter. *Laboratorní scénáře popisující systém DNS*. Brno, 2018, 76 s. Diplomová práca. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedúci práce: doc. Ing. Jan Jeřábek, Ph.D.

VYHLÁSENIE

Vyhlasujem, že som svoju diplomovoú prácu na tému „Laboratorní scénáře popisující systém DNS“ vypracoval(a) samostatne pod vedením vedúceho diplomovej práce, využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor(ka) uvedenej diplomovej práce ďalej vyhlasujem, že v súvislosti s vytvorením tejto diplomovej práce som neporušil(a) autorské práva tretích osôb, najmä som nezasiahol(-la) nedovoleným spôsobom do cudzích autorských práv osobnostných a/alebo majetkových a som si plne vedomý(-á) následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona Českej republiky č. 121/2000 Sb., o práve autorskom, o právach súvisiacich s právom autorským a o zmene niektorých zákonov (autorský zákon), v znení neskorších predpisov, vrátane možných trestnoprávnych dôsledkov vyplývajúcich z ustanovenia časti druhej, hlavy VI. diel 4 Trestného zákoníka Českej republiky č. 40/2009 Sb.

Brno ....

podpis autora(-ky)

POĎAKOVANIE

Rád by som poďakoval vedúcemu diplomovej práce pánovi doc. Ing. Janu Jeřábkovi, Ph.D. za odborné vedenie, konzultácie, trpezlivosť a konštruktívne návrhy k práci.

Brno ....

podpis autora(-ky)

Faculty of Electrical Engineering
and Communication
Brno University of Technology
Purkynova 118, CZ-61200 Brno
Czech Republic
http://www.six.feec.vutbr.cz

POĎAKOVANIE

Výzkum popsaný v tejto diplomovej práci bol realizovaný v laboratóriách podporených projektom SIX; registračné číslo CZ.1.05/2.1.00/03.0072, operačný program Výzkum a vývoj pro inovace.

Brno ....

podpis autora(-ky)

EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI

OP Výzkum a vývoj
pro inovace

Tato práce vznikla jako součást klíčové aktivity KA6 - Individuální výuka a zapojení studentů bakalářských a magisterských studijních programů do výzkumu v rámci projektu OP VVV Vytvoření double-degree doktorského studijního programu Elektronika a informační technologie a vytvoření doktorského studijního programu Informační bezpečnost, reg. č. CZ.02.2.69/0.0/0.0/16_018/0002575.

EVROPSKÁ UNIE
Evropské strukturální a investiční fondy
Operační program Výzkum, vývoj a vzdělávání

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

Projekt je spolufinancován Evropskou unií.

OBSAH

ZOZNAM OBRÁZKOV

ZOZNAM TABULIEK

ZOZNAM VÝPISOV

ÚVOD

Systém doménových mien (DNS) je pre ľudí v súčasnosti samozrejmosťou a mnohí si ani nevedia predstaviť, ako by to v dnešnej internetovej dobe vyzeralo bez tohto veľmi užitočného „nástroja“. Systém DNS začal vznikať ešte pred érou internetu, tak ako ho poznáme dnes. Bolo to už za čias ARPANET-u, kedy sa začali používať názvy mapované na adresy. To je podstata systému DNS dodnes. Nemusíme si pamätať enormné množstvo IP adries serverov, ktoré navštevujeme denne, a dokonca keď sa ešte tieto adresy tak často menia. O začínajúcom rozmachu IPv6 nehovoriac. Za systémom DNS dnes stojí mnoho medzinárodných organizácií, ktoré okrem iného, prevádzkujú 13 koreňových serverov po celom svete, bez ktorých by tento systém nemohol fungovať. V skutočnosti ich je ale omnoho viac. To len odzrkadľuje dôležitosť systému DNS.

Táto práca sa v úvode venuje práve systému DNS, jeho štruktúre, hierarchii doménových mien a ďalším veciam, ktoré sú so systémom DNS úzko späté. Práca rozoberá súčasné možnosti DNS a popisuje najpopulárnejšie implementácie.

Ďalšia časť je venovaná rôznym technológiám, ktoré by mohli napomôcť študentovi komunikačných technológií ľahko pochopiť spôsob, akým tento systém funguje. K tomu určite prispieva dobre navrhnuté laboratórne prostredie, aj keď iba virtuálne.

Záver práce je venovaný realizácii samotných laboratórnych úloh, vrátane postupu pre študenta a stručný popis možného riešenia.

1 SYSTÉM DNS

Všetky počítače, ktoré spolu komunikujú prostredníctvom internetu, používajú na svoju identifikáciu IP adresu (IPv4 a v súčasnosti aj čoraz viac používanú IPv6). Pre človeka sú takéto adresy len ťažko zapamätateľné, a preto bol vytvorený systém doménových mien (*Domain Name System* - **DNS**).[1, s. 255]

1.1 Štruktúra doménových názvov

Systém DNS nám teda umožňuje používať tzv. doménové názvy namiesto IP adries všade, kde máme prístup k databáze DNS. Doménou označujeme skupinu počítačov, ktoré sú v určitom vzťahu (napr. tvoria organizačnú jednotku alebo zdieľajú geografickú polohu). Táto databáza je decentralizovaná, má stromovú štruktúru a informácie o jednotlivých doménach poskytujú doménové servery.[1, s. 255]

Doménové meno sa skladá z reťazcov, ktoré sú od seba oddelené bodkou. Koreň tejto štruktúry sa v DNS nazýva koreňová doména (*root domain*) a označuje sa bodkou na konci doménového mena, ktorú pri bežnom používaní písať nemusíme. Pod koreňovou doménou sa nachádzajú domény najvyššej úrovne (*Top Level Domain* - **TLD**). Tieto môžeme podľa druhu rozdeliť nasledovne:[2, s. 60]

- **národné** (ccTLD), ktoré majú 2 znaky a označujú kód krajiny podľa normy ISO-3166: cz, sk, eu... [1, s. 256];
- **internacionalizované národné** (IDN ccTLD): ею, ελ... [3];
- **generické** (gTLD): com, org, edu, digital, online[1]... [1, s. 256];
- **infraštruktúrne**: arpa[4];
- **rezervované** na špeciálne účely: example, invalid, localhost, local, onion, test[5].

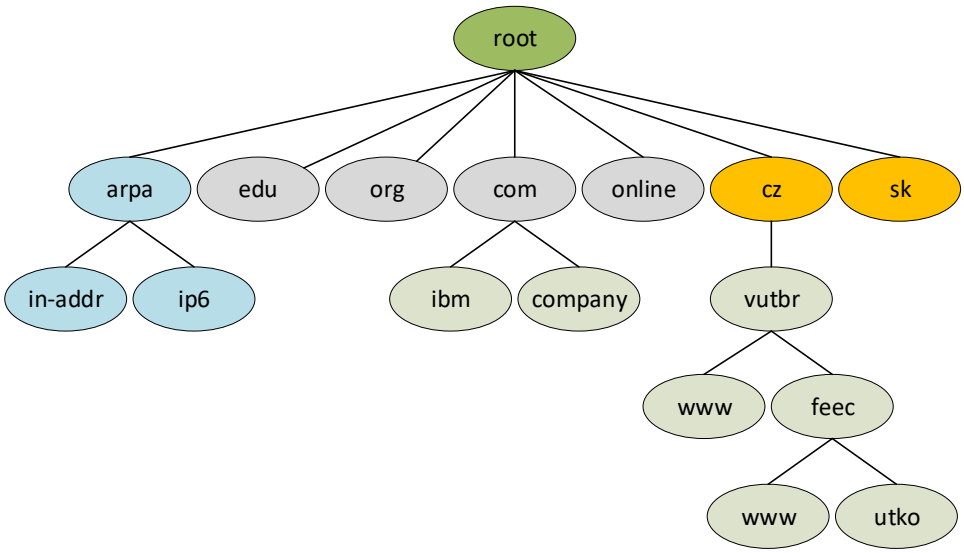
TLD sa potom delí na domény druhej úrovne (skrátene len domény), ktoré môžu vlastniť jednotlivé organizácie. Príkladom takejto domény je `vutbr.cz`. Doména sa môže ďalej deliť na subdomény rôznej úrovne, napr. `www.vutbr.cz` alebo `utko.feec.vutbr.cz`. Tento príklad a znázornenie stromovej štruktúry v DNS môžeme vidieť na obrázku 1.1.[2] Doménový názov môže mať celkovo maximálne 255 znakov, kde reťazce na jednotlivých úrovniach môžu mať maximálne 63 znakov.[2, s. 60]

FQDN (*Fully Qualified Domain Name*) je označenie pre úplne špecifikované doménové meno, ako napríklad adresa `www.utko.feec.vutbr.cz.`, ktorá špecifikuje názov konkrétneho počítača (webový server). V prípade, že by sme sa nachádzali

[1] Organizácia ICANN spustila v júni 2011 „New gTLD Program“, ktorý umožnil registrovanie nových generických TLD[6].

[2] Toto znázornenie neobsahuje všetky existujúce domény, slúži len ako ukážka.

v sieti patriacej pod doménu `utko.feec.vutbr.cz.`, tak by sme mohli použiť aj Non-FQDN, v tomto prípade `www` (bez bodky na konci).[1, s. 257]



Obr. 1.1: Názvy domén tvoriace stromovú štruktúru v systéme DNS

IDN (*Internationalized Domain Names*) označuje systém pre podporu diakritiky a rôznych národných znakov. Umožňuje používať v názvoch domén znaky, ktoré nepatria do tabuľky **ASCII**, ako napr. azbuka, grécke alebo čínske znaky atď. Príkladom je pravdepodobne jediná česká doména `www.háčkyčárky.cz`[7]. Aby sa kvôli podpore IDN nemusel prerábať celý systém DNS, reťazec sa so znakmi, ktoré nepatria do ASCII, prevedie do formátu **ACE** (*ASCII Compatible Encoding*) pomocou tzv. *Punycode* a ten je možné uložiť do DNS bez ďalších zásahov v systéme. Vyššie spomenutá doména sa tak prevedie na `www.xn--hkyrky-ptac70bc.cz` kde `xn--` je prefix, podľa ktorého sa dá rozoznať, že bol vykonaný prevod do ACE.[2, s. 66-67]

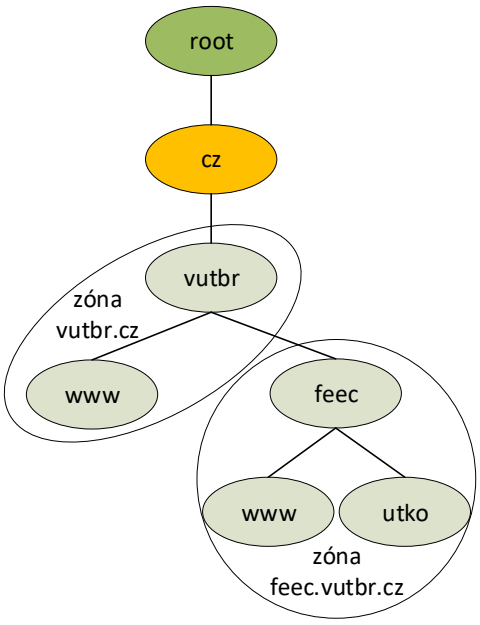
1.2 Hierarchia doménových serverov

Zóna je definovaná ako autoritatívny zdroj informácií o všetkých názvoch domén, ktoré sú do nej začlenené. Jedna zóna má vždy jeden primárny a ľubovoľný počet sekundárnych doménových serverov. Príklad je znázornený na obrázku 1.2, kde môžeme vidieť dve zóny. Toto ale neplatí v root zóne, kde sú všetky servery (13 serverov, označené písmenami A až M) primárne. Doménový server uchováva informácie (autoritatívne alebo cache) v tzv. *zónových súboroch*. **Primárny** server po spustení načíta do pamäte autoritatívne informácie z lokálneho úložiska. **Sekundárne** servery načítajú tieto informácie (skopírujú) priamo z primárneho servera

pomocou tzv. zónových prenosov (*zone transfer*). Toto sa potom opakuje vždy, keď nastane na primárnom serveri nejaká zmena.[2, s. 63]

Rekurzívny DNS server s ukladaním do pamäte cache nemá staticky uložené záznamy vo svojej pamäti. Keď dostane požiadavku o nejakú informáciu z DNS, snaží sa získať túto informáciu od ostatných autoritatívnych serverov. Aby sa v prípade opakovaných požiadaviek nemusel dopytovať znova, ukladá si zistené informácie do cache pamäte (platnosť týchto dát je obmedzená údajom v poli TTL). Samotný DNS server môže byť zároveň autoritatívny aj rekurzívny resolver. V tomto prípade, ak je požiadavka na informáciu v rámci zóny, pre ktorú je autoritatívny, pošle autoritatívnu odpoveď. V prípade že sa ale jedná o požiadavku mimo tejto zóny alebo zón, zisťuje odpoveď od iných DNS serverov. V takomto prípade je odpoveď od rekurzívneho servera neautoritatívna (daný server sa za ňu nemôže zaručiť). Rekurzívne DNS servery nasadzujú napríklad poskytovatelia internetových služieb alebo väčšie spoločnosti.[1, s. 268]

Stub resolver síce nepatrí medzi doménové servery, bezprostredne sa ale zúčastňuje DNS prekladu, keďže je súčasťou OS. Stub resolver umožňuje, aby naše počítače mohli pracovať so systémom DNS a komunikovať s doménovými servermi na internete.[2, s. 62]



Obr. 1.2: Zóny v systéme DNS

1.3 Typy záznamov v DNS (vety RR)

Informácie o doménových menách, či už ide o príslušné IP adresy, ale aj všetky ostatné informácie, sú uložené v pamäti DNS serverov v tvare zdrojových viet (*Resource Records* - **RR**). Tieto vety obsahujú (v úplnom tvare) niekoľko polí, ktoré sú znázornené na obrázku 1.3. Nasledujúci zoznam stručne popisuje základné a najčastejšie používané typy viet.[1, 2]

- **SOA** (*Start of Authority*) - záznam je práve jeden vždy na začiatku konfiguračného súboru zóny a určuje menný server, ktorý je autoritatívnym zdrojom informácii pre danú doménu.
- **A** (*A host address*) - záznam poskytuje mapovanie doménového mena na 32 bitovú IPv4 adresu.
- **AAAA** (*IP6 address*) - podobne ako A záznam, ale používa 128 bitové IPv6 adresy.
- **MX** (*Mail eXchange*) - obsahuje dve informácie: 16 bitové číslo ako preferenciu a doménové meno e-mailového servera.
- **CNAME** (*Canonical name for an alias*) - doménové meno, ktoré špecifikuje alias (synonymum) k inému menu. Môže odkazovať iba na A/AAAA záznam, nie iný CNAME.
- **NS** (*Authoritative name server*) - definuje autoritatívne doménové servery pre danú doménu. Musí odkazovať na doménové meno, ktorému je priradená IP adresa pomocou A/AAAA záznamu (podobne ako pri CNAME).
- **PTR** (*Domain name pointer*) - slúži k prekladu IP adresy na doménové meno (reverzný preklad), tj. k prekladu prvkov domény `in-addr.arpa` (IPv4) prípadne `ip6.arpa` (IPv6).
- **TXT** (*Text string*) - obsahuje vo svojej dátovej časti všeobecný textový reťazec.
- **SRV** (*Service locator*) - má za úlohu poskytovať k názvom počítačov taktiež názvy služieb na konkrétnej doméne. Tieto záznamy využívajú napr. systémy Microsoftu (LDAP).

Obr. 1.3: Zloženie úplnej vety RR (Resource Record)

1.4 Protokol DNS

Protokol DNS je typu klient-server, pracuje na 4. vrstve modelu TCP/IP a ako transportný protokol využíva UDP aj TCP, oboje na **porte 53**. Protokol DNS de-

finuje niekoľko typov operácií, pričom dva základné sú **DNS QUERY** (*žiadosť*) a **DNS QUERY RESPONSE** (*odpoveď na žiadosť*). Ako už ich názov prezrádza, prvý typ správy slúži na dopytovanie sa na preklad doménového mena, ten druhý je použitý na následnú odpoveď, ktorá v prípade úspešného prekladu obsahuje požadovanú informáciu. Oba tieto operácie používajú spoločný **formát** správ a je zložený z nasledujúcich častí:[2, s. 66]

- **HEADER** (*záhlavie*) - obsahuje základné informácie danej správy, napr. identifikačné číslo (rovnaké pre žiadosť aj odpoveď), typ správy (query/response), autoritatívnosť odpovede, požadovanie/umožnenie rekurzie, návratový kód, autenticita a iné. Záhlavie je jediná povinná časť DNS správy a v ňom je definovaný výskyt nasledujúcich častí správy (QUERY, ANSWER...) a ich počet.
- **QUERY** (*požiadavka*) - neúplné vety RR obsahujúce iba požadované meno na preklad (*Query Name*), typ vety (*Type*) a triedu (*Class*).
- **ANSWER** (*odpoveď*) - úplné vety RR vrátane odpovede na predchádzajúce požiadavky.
- **AUTHORITY** (*autorita*) - použité v prípade viet typu NS, kde sa nachádzajú informácie o autoritatívnych serveroch požadovanej domény.
- **ADDITIONAL** (*doplňujúce informácie*) - táto časť často obsahuje IP adresy DNS serverov, ktoré boli vrátené v časti AUTHORITY a zdrojový DNS server mal k dispozícii aj ich IP adresy. Táto časť sa používa aj na prenos informácií o DNSSEC.

Medzi ďalšie operácie patrí **DNS NOTIFY** (*oznámenie*), ktorá slúži primárnemu serveru na informovanie sekundárnych serverov o zmene v zónovom súbore. Používa sa na urýchlenie aktualizácie zónových súborov medzi servermi, pričom sa najčastejšie používa technika inkrementálneho zónového prenosu. Pre zónové prenosy sa využíva výlučne TCP protokol kvôli spoľahlivosti. **DNS UPDATE** (*aktualizácia*) sa využíva na vzdialené dynamické úpravy záznamov v databáze (použité napríklad v *MS Active Directory*).[2, s. 66]

1.4.1 Preklad doménových názvov

Na preklad doménových názvov na IP adresy sa používajú dve metódy:

- **iteratívna** (*iterative*),
- **rekurzívna** (*recursive*).

Pomocou **iteratívnej** metódy sa doménový server dopytuje iných serverov, keď chce získať autoritatívne informácie o hľadanom doménovom názve. Najprv sa spýta root servera, ten mu poskytne iba adresu doménového servera druhej úrovne, ktorý by to mohol vedieť. Dopytujúci server sa teda spýta aj tohto servera, no ten mu odpovie

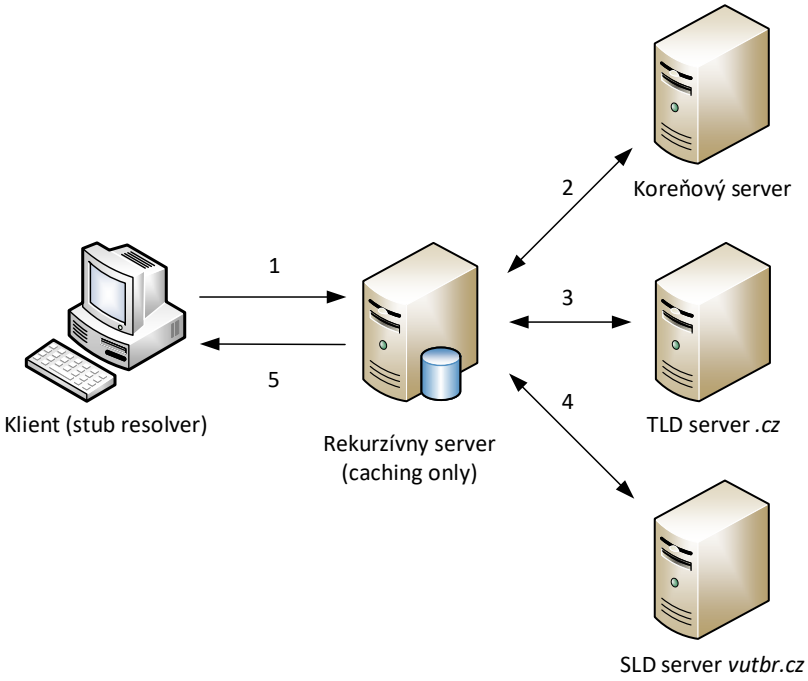
podobne a odkáže ho na iný server. Takto to pokračuje, až kým neodpovie DNS server, ktorý je autoritatívny pre cielenú doménu a odpovie mu na požiadavku. Rekurzívna metóda spočíva v tom, že ak stub resolver (v klientskom PC) potrebuje preložiť nejaký doménový názov, požiadavku pošle na rekurzívny resolver a ten sa postará o samotné hľadanie, napríklad iteratívnou metódou (ak nie je nastavený len ako DNS forwarder). Konečnú odpoveď potom odošle stub resolveru.[1, s. 271]

Na obrázku 1.4 je znázornený príklad, ktorý opisuje využitie oboch vyššie spomenutých metód prekladu domén. V modelovej situácii chce klient získať IP adresu webového servera `www.vutbr.cz` a vo svojej konfigurácii má nastavené, aby používal zobrazený rekurzívny server s pamäťou cache. Predpokladáme, že v tejto pamäti sa aktuálne nenachádza žiaden záznam. Hneď po tom, čo klient vyčerpá všetky lokálne prostriedky na preklad, začne kontaktovať rekurzívny server (bod 1). Ten sa následne pozrie do svojej cache pamäte, či v nej nenájde požadovaný záznam. Záznam nenájde, takže začne iteratívnou metódou zisťovať informácie o danej adrese. Skontaktuje koreňový server (jeho adresy má uložené lokálne) a spýta sa ho na adresu `www.vutbr.cz` (bod 2). Koreňový server mu odpovie, ale v odpovedi je len adresa (vrátane IP) DNS servera, ktorý by mu mal poskytnúť presnejšie informácie. Kontaktuje teda DNS server domény najvyššej úrovne `.cz` a opýta sa ho opäť na adresu `www.vutbr.cz` (bod 3). Rovnako ako v predchádzajúcom bode, získa adresu doménového servera, tento krát to je server druhej úrovne `vutbr.cz`. Kontaktuje nakoniec aj tento server, ktorý mu odpovie (autoritatívne), už s požadovanou IP adresou servera `www.vutbr.cz` (bod 4). Rekurzívny server túto adresu odošle klientovi ako odpoveď na jeho žiadosť (bod 5) a nakoniec si všetky získané informácie odloží do pamäte cache, kde budú uchované počas doby platnosti podľa hodnoty TTL konkrétnych záznamov.[2, s. 65]

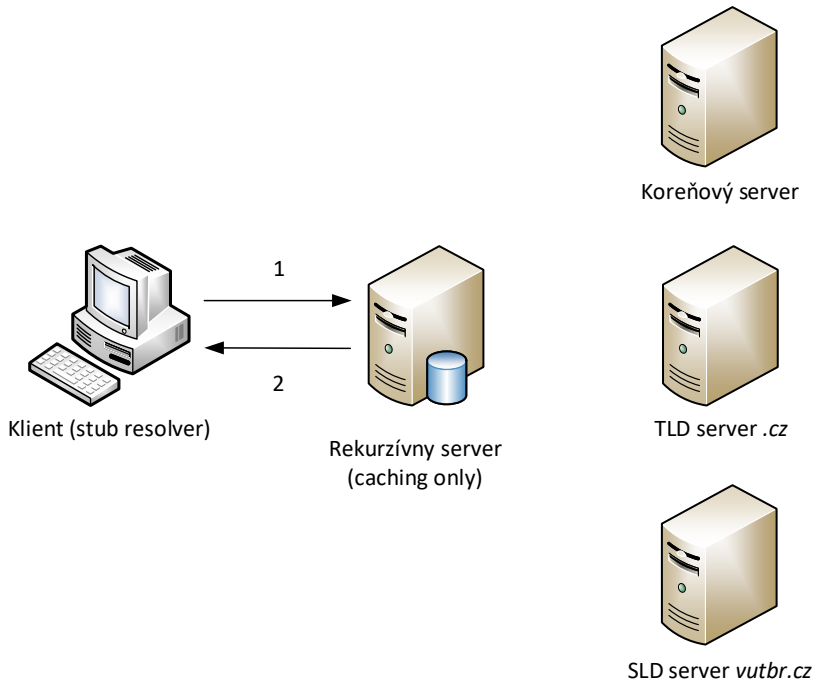
Obrázok 1.5 znázorňuje rovnakú situáciu, avšak s jedinou zmenou – rekurzívny server už má potrebné dáta (platné) na preklad v svojej pamäti cache. Ak teda klient (nemusí byť ten istý) požiada o preklad adresy `www.vutbr.cz` (bod 1), rovno mu pošle odpoveď obsahujúcu hľadanú IP adresu (bod 2). V tomto prípade je preklad oveľa rýchlejší.[2, s. 65]

1.4.2 Reverzný preklad

Okrem klasického prekladu doménového názvu na IP adresu, vyžadujú niektoré aplikácie aj opačný preklad. Ten nazývame ako reverzný a využívajú sa pri ňom tzv. reverzné domény. Tieto domény patria pod infraštruktúrnu pseudodoménu `arpa.` a sú to `in-addr.arpa.` pre IPv4 a `ip6.arpa.` pre IPv6. Príkladom reverznej domény môže byť `37.47.195.in-addr.arpa.` do ktorej patrí sieť `195.47.37.0/24` alebo `18.37.47.195.in-addr.arpa.` do ktorej patrí koncový uzol s adresou `195.47.37.18/`



Obr. 1.4: Doménový preklad (pamäť cache neobsahuje hľadaný záznam)



Obr. 1.5: Doménový preklad (pamäť cache obsahuje hľadaný záznam)

32. V IPv6 by adresa `2001:8b0:c5:1::1` patrila do domény `1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0.0.0.5.c.0.0.0.b.8.0.1.0.0.2.ip6.arpa`. Keďže IP adresy tvoria taktiež stromovú štruktúru, nie je zložité tieto adresy integrovať do DNS. Toto sa uskutočňuje pomocou záznamov PTR, ktoré mapujú reverznú doménu na bežný doménový názov, napr. `server1.vutbr.cz`.[1, s. 258]

Reverzný preklad je v súčasnosti využívaný napríklad pri overovaní komunikácie dvoch SMTP serverov počas posielania elektronickej pošty. Prijímajúci server si zistí doménové meno pomocou reverzného prekladu podľa IP adresy servera. Potom toto meno použije pri preklade na IP adresu, a ak sa pôvodná a preložená IP zhoduje, považuje sa tento server za overený.[1, s. 258]

1.5 DNSSEC

DNSSEC (*Domain Name System Security Extensions*) je rozšírenie systému DNS, pomocou ktorého vieme zabezpečiť pravosť získaných záznamov.[8] Funguje na princípe podpisovania záznamov. Nejde teda o šifrovanie samotných záznamov. Rekurzívny resolver v prípade zabezpečenej domény má možnosť overiť autenticitu získaných informácií. Toto zabezpečenie teda predpokladá bezpečný kanál medzi rekurzívnym resolverom a klientom. Je tak vhodné používať resolver, ktorý je umiestnený v lokálnej sieti, alebo ešte lepšie priamo v klientovi. DNSSEC pridáva nové typy zdrojových viet (RR):[2, s. 68]

- **DNSKEY** (*DNS Key record*) - obsahuje verejný kľúč, ktorý slúži na overovanie podpisov.
- **RRSIG** (*Resource Record Signature*) - obsahuje podpis pre konkrétny záznam. Tento podpis je overovaný resolvermi pomocou verejného kľúča v zázname DNSKEY.
- **DS** (*Delegation Signer*) - odtlačok verejného kľúča, ktorý sa vkladá do nadradenej domény, kde je podpísaný kľúčom nadradenej domény.
- **NSEC** (*Next Secure record*) - obsahuje odkaz na najbližší záznam (z pohľadu abecedného poradia), ktorý sa využíva v prípade odpovede na neexistujúci záznam (NXDOMAIN).
- **NSEC3** (*Next Secure record version 3*) - vylepšuje NSEC, keď namiesto reálneho mena obsahuje iba jeho hash. To zabraňuje prehľadávaniu celej zóny, čo umožňovali práve NSEC záznamy.

DNSSEC využíva pri podpisovaní asymetrickú kryptografiu, tzn. že využíva privátny kľúč na podpisovanie záznamov a verejný kľúč na validovanie podpisu. V súčasnosti sa využívajú dva typy kľúčov, a to KSK (*key signing keys*) a ZSK (*zone signing keys*). KSK bývajú väčšie a slúžia na podpis ostatných kľúčov a teda aj ZSK,

ktoré sú spravidla menšie, a slúžia na podpis ostatných záznamov. Týmto rozdelením sa zjednodušuje výmena kľúčov (KSK nie je potrebné meniť tak často), a tiež týmto redukujeme veľkosť RRSIG/DNSKEY záznamov.[10]

Základom overenia je reťazec dôvery, ktorý sa skladá z prepojených DS a DNSKEY záznamov, začínajúci vždy v koreňovej doméne, ktorej bod dôvery (trust anchor) je obsiahnutý v DNS serveroch ako súčasť programu, prípadne sa dá stiahnuť zo stránok organizácie IANA[9]. DNSSEC validácia je možná len vtedy, keď je tento reťazec kompletný až po dopytovanú doménu.

2 IMPLEMENTÁCIE PROTOKOLU DNS

2.1 Autoritatívne DNS servery

2.1.1 BIND

Medzi najznámejšie a najviac používané softvérové implementácie DNS protokolu patrí jednoznačne **BIND** (*Berkeley Internet Name Domain*), ktorý vznikol v osemdesiatych rokoch na Kalifornskej univerzite v Berkeley a v súčasnosti je vyvíjaný konzorciom **ISC** (*Internet Software Consortium*). BIND, niekedy označovaný ako `named`, je dostupný v podobe zdrojového kódu pre Unix-like operačné systémy, kde sa BIND považuje za de facto štandard, a takisto pre Windows v podobe spustiteľných súborov. BIND je najviac používaný autoritatívny server v koreňovej zóne. Posledná verzia 9 bola vytvorená už v roku 2000, aktuálne to je 9.11.2.[11]

BIND je open-source softvérový balík zložený z týchto častí:

- doménový resolver,
- autoritatívny DNS server,
- nástroje ako napr. `dig`, `nslookup`, `host` atď.

Resolver sa stará o získavanie informácií od vzdialených autoritatívnych serverov. Tento resolver (použitý ako rekurzívny server) potom preposiela odpoveď stub resolverom, ktoré sú súčasťou OS. Knižnica BIND resolveru býva taktiež často použitá ako stub resolver v mnohých distribúciach OS Linux. **Autoritatívny** DNS server naopak odpovedá na žiadosti od resolverov a poskytuje im informácie, ktoré sú pre daný server autoritatívne. **Nástroje** slúžia administrátorom na uľahčenie správy DNS serverov, použiteľné všeobecne v systéme DNS.

BIND ukladá dáta a nastavenia v týchto umiestneniach:

- `/etc/bind/` - konfiguračné súbory, ktoré sa dlhodobo nemenia,
- `/var/lib/bind/` - zónové súbory primárnych zón,
- `/var/cache/bind/` - zónové súbory sekundárnych zón, prípadne automaticky podpisované primárne zóny (DNSSEC),
- pamäť RAM - vyrovnávacia pamať (cache).

2.1.2 NSD

NSD (*Name Server Daemon*) je, podobne ako BIND, open-source DNS server, ktorý vyvíja *NLnet Labs* od roku 2002. Na rozdiel od servera BIND, bol NSD server vytvorený ako rýdzo autoritatívny DNS server, ktorý má poskytovať vysoký výkon v spojení s jednoduchosťou. Zónové súbory zo servera BIND môžu byť jednoducho použité aj v NSD. NSD sa často používa ako autoritatívny server v root zóne.

Posledná stabilná verzia NSD je 4.1.19.[12]

2.1.3 Knot DNS

Knot DNS je ďalší veľmi známy výlučne autoritatívny DNS server s otvoreným zdrojovým kódom. Vyvíja ho združenie *CZ.NIC*, ktoré okrem iného, prevádzkuje českú národnú doménu `.cz`. Medzi jeho výhody patrí vysoký výkon, dobrá škálovateľnosť a podpora všetkých hlavných funkcií protokolu DNS. Vývoj začal v roku 2011 a jeho hlavným motívom bol malý počet slobodných implementácií autoritatívnych DNS serverov vhodných na prevádzku TLD domény (v tom čase použiteľné iba BIND a NSD). Táto implementácia je taktiež použitá v root zóne na koreňových serveroch „K“ (RIPE NCC)[13] a „L“ (ICANN)[14]. V súčasnosti je k dispozícii posledná stabilná verzia 2.6.3.[15]

2.2 Rekurzívne DNS servery

2.2.1 Unbound

NLnet Labs od roku 2006 vyvíja okrem NSD aj implementáciu servera **Unbound** v jazyku C. Tento je rekurzívny DNS server, ktorý je rovnako open-source, podporuje ukladanie do cache pamäte a validovanie pomocou DNSSEC. Vývoj pôvodnej verzie (v jazyku Java) začal už v roku 2004 pod záštitou *Verisign labs*, *Nominet*, *Kirei* a *ep.net*. Unbound je priamou konkurenciou servera BIND v roli rekurzívneho resolvera. Bol postavený ako modulárny softvér, ktorého časti tvoria jednotlivé funkcie systému DNS. V súčasnosti nájdeme Unbound vo väčšine distribúcií, vrátane tých pre embedded zariadenia (OpenWRT, LEDE). Momentálne posledná stabilná verzia je 1.6.7.[16]

2.2.2 Knot Resolver

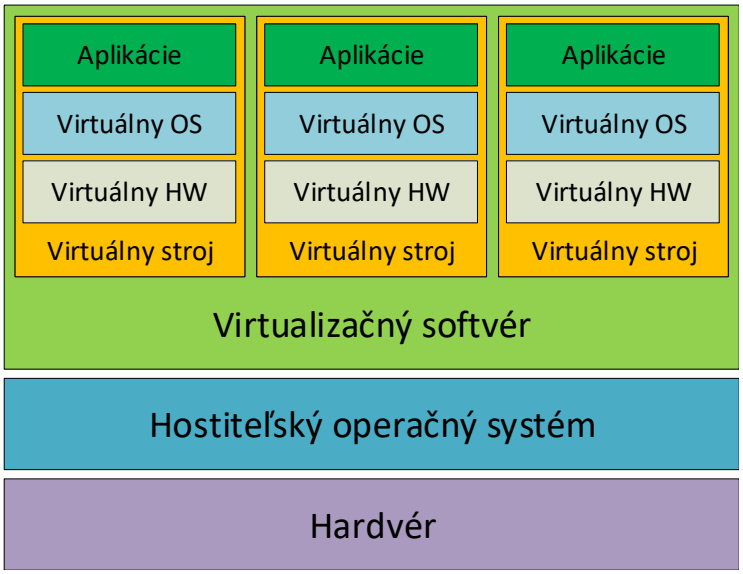
Knot Resolver je, ako už názov napovedá, vyvíjaný rovnakým združením ako Knot DNS a to *CZ.NIC*. Knot Resolver je rýchly a flexibilný rekurzívny DNS server. Medzi jeho prednosti patria dynamická konfigurácia v Jazyku Lua, malé a výkonné jadro zabezpečujúce DNS a DNSSEC z projektu Knot DNS a taktiež veľmi dobrá rozšíriteľnosť pomocou dynamických modulov v jazyku C, Lua alebo Go. Tento server podporuje okrem iného aj najnovšie štandardy, ako napríklad *DNS over TLS* alebo *DNS Cookies*. Aktuálne je Knot Resolver v stabilnej verzii 1.5.1.[17]

3 POUŽITÉ TECHNOLÓGIE A SOFTVÉR

V laboratórnych úlohách sú využité niektoré softvérové nástroje a technológie umožňujúce vykonanie daných úloh, prípadne uľahahčujúce prácu v laboratórnom prostredí. Tieto sú stručne popísané v nasledujúcom texte.

3.1 Virtualizačná platforma Oracle VM VirtualBox

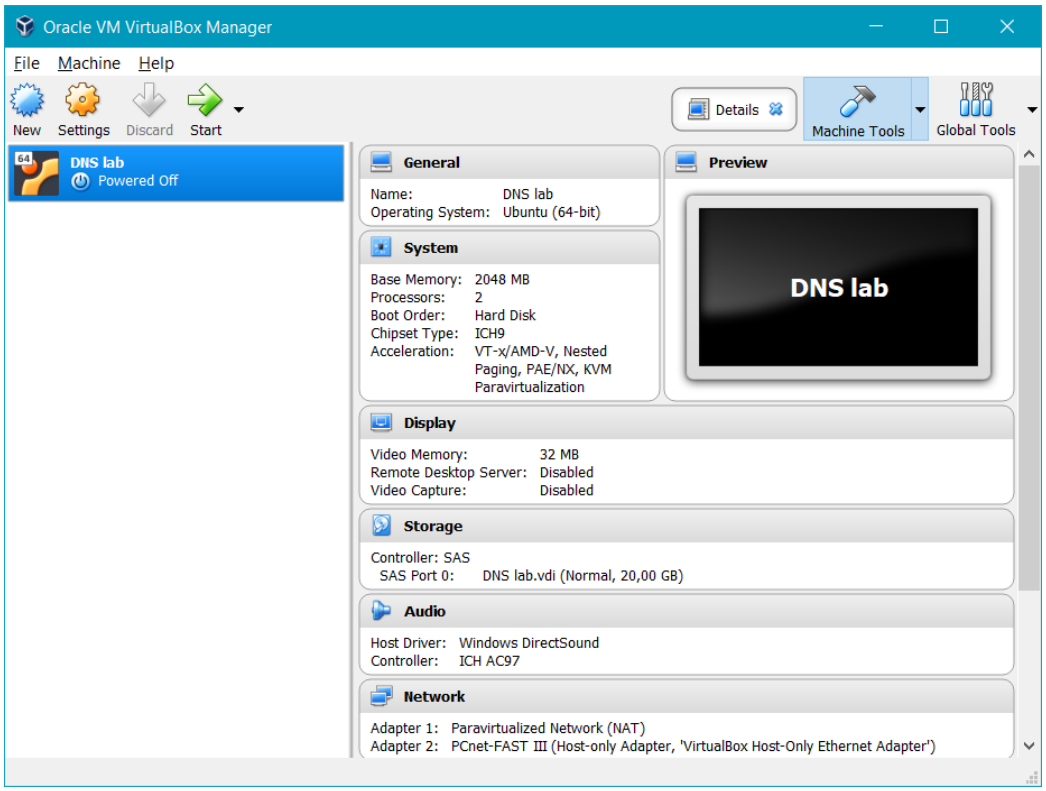
VirtualBox je jeden z najznámejších open-source virtualizačných nástrojov, ktorého vývoj je aktuálne spravovaný spoločnosťou Oracle (pod licenciou GPL). Tento virtualizačný nástroj umožňuje spustiť ľubovoľný operačný systém (guest) pod naším hostiteľským operačným systémom (host). Ako hostiteľské operačné systémy podporuje Windows, Linux, Mac OS X alebo Solaris. Virtuálne stroje (virtual machines) podporujú OS Windows, rôzne Linuxové distribúcie, UNIXové operačné systémy ako aj operačné systémy typu Solaris a iné.[20]



Obr. 3.1: Virtuálne stroje bežiace na spoločnom hardvéri

Sieťové pripojenie môžeme riešiť viacerými spôsobmi. Prvou z možností je NAT medzi hostiteľským OS a virtuálnym strojom. To umožňuje sprístupniť internetovú konektivitu virtuálnemu stroju bez ďalších zložitých nastavovaní (v prípade že má hostiteľský OS prístup k internetu). Druhá možnosť je sieťový most (bridge), ktorý nám umožní pripojiť sa priamo do sietí, ktoré sú prístupné hostiteľskému OS. Prakticky budú takto prepojené virtuálny stroj a hostiteľský OS s vonkajšou sieťou

cez virtuálny switch. Ďalšou možnosťou je pripojenie výlučne k hostiteľskému OS pomocou tzv. host only virtuálneho sieťového adaptéra.[20]

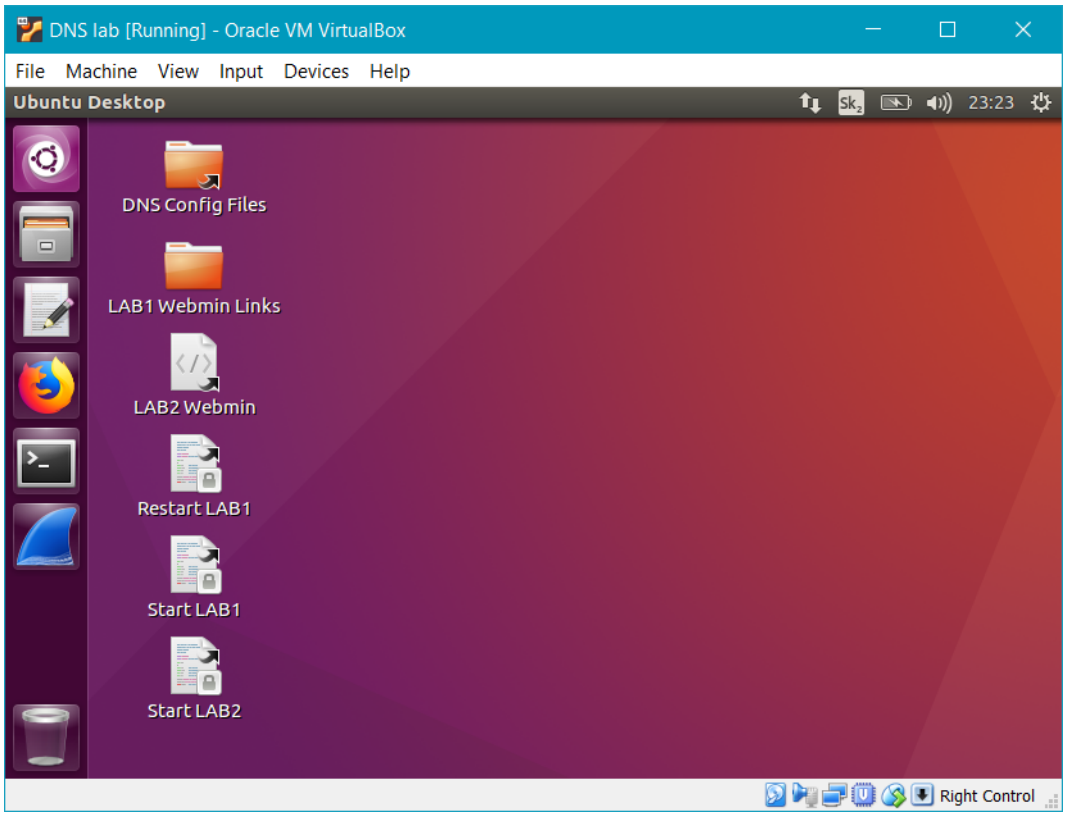


Obr. 3.2: Základné okno Oracle VM VirtualBox

3.2 Ubuntu

Ubuntu je jedna z najpoužívanejších distribúcií pre pracovné stanice (desktopy) založená na linuxovej distribúcii Debian. Ubuntu ale nie je len vo verzii pre desktopy (v rôznych deriváciách podľa použitého grafického prostredia, napr. Unity, GNOME, MATE, KDE...), ale aj pre servery (bežne bez GUI), či zariadenia s dotykovým displejom. Taktiež existuje verzia pre použitie v oblasti cloud computingu. Všetky tieto odnože Ubuntu používajú ten istý základ systému, líšia sa iba nainštalovaným softvérom.[18]

Tento projekt je sponzorovaný spoločnosťou Canonical. Verzie s dlhodobou podporou (*Long Term Support* - LTS) vychádzajú pravidelne každý druhý rok. V čase písania tohto textu bola aktuálna verzia s dlhodobou podporou **Ubuntu 18.04 LTS „Bionic Beaver“**.[19] V tejto práci je využitá staršia verzia Ubuntu 16.04.4 LTS podporovaná do roku 2021 z dôvodu lepšej podpory využívaného softvéru, ako aj neskorého dátumu vydania novšej verzie.



Obr. 3.3: Spustený Ubuntu Desktop 16.04 v Oracle VM VirtualBox

3.3 Docker

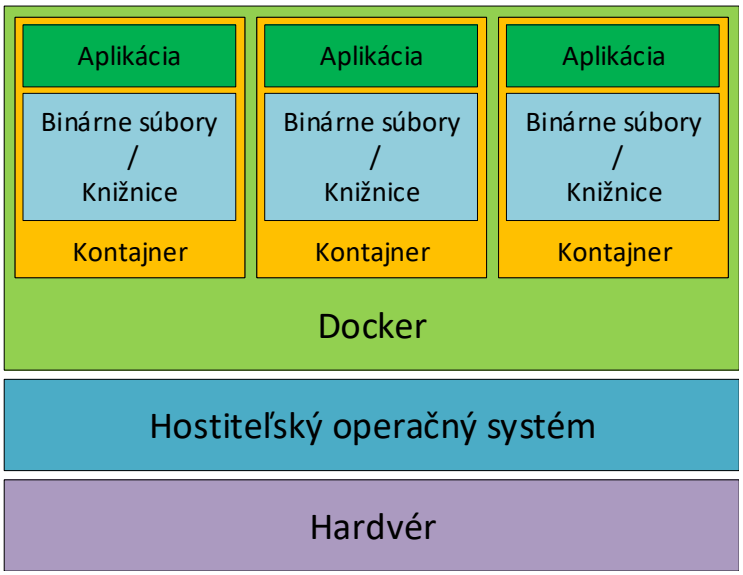
Docker je open-source nástroj vyvíjaný spoločnosťou Docker, Inc, ktorý poskytuje jednotné rozhranie pre izoláciu aplikácii v prostredí Linuxu aj Windowsu, čím tieto aplikácie oddeľujú od infraštruktúry na ktorej bežia. Toto dovoľuje sprístupňovať softvér oveľa rýchlejšie a jednoduchšie.[21]

Docker vo svojom princípe využíva virtualizáciu na úrovni operačného systému čo umožňuje na rozdiel od klasickej virtualizácie spustiť viacej na sebe nezávislých aplikácii z dôvodu oveľa nižšej réžii. Tieto aplikácie bežia v tzv. kontajneroch, ktoré využívajú spoločné jadro operačného systému, pričom sú vzájomne nezávislé a izolované. Docker sa skladá z nasledujúcich častí:[21]

- client,
- host:
  - daemon,
  - images,
  - containers,
- registries.

Docker využíva architektúru klient-server. Docker Daemon (alebo server) je zod-

povedný za činnosti spojené s kontajnermi. Daemon prijíma príkazy od Docker Client cez CLI alebo REST API. Docker Client môže byť spustený na rovnakom hostiteľovi ako Daemon alebo na vzdialenom.



Obr. 3.4: Beh aplikácií v kontajneroch

Obraz (image) je základný stavebný prvok Dockera. Je to šablóna (template) pre vytvorenie kontajnera, ktorý spravidla obsahuje nejaký softvér a môže byť postavený na inom obraze (base). Jednotlivé kontajnery sú vytvorené pomocou obrazov, kde jeden obraz môže byť súčasne použitý v rôznych kontajneroch. Docker registre slúžia ako repozitáre obrazov, pričom sa v základe využíva oficiálny Docker Hub. V prípade, že chceme spustiť nejaký softvér v kontajneri, máme dve možnosti. Ak existuje obraz daného softvéru v prístupnom registri a vyhovuje nám (dôveryhodnosť), tak ho stačí stiahnuť a použiť. V opačnom prípade si musíme obraz vytvoriť.

Ako príklad sa dá použiť spustenie webového servera v kontajneri. Je to výhodné najmä vtedy, keď potrebujeme otestovať rôzne konfigurácie alebo spustiť viacero inštancií tohto serveru. V základnej inštalácii Dockeru tak pre stiahnutie obrazu napr. pre Apache stačí spustiť príkaz „`docker pull httpd`“, ktorý vyhľadá obraz s názvom `httpd` (Apache) a stiahne ho do lokálneho registra. V ďalšom kroku stačí spustiť web server s niekoľkými parametrami: „`docker run -dit -name my-apache-app -p 8080:80 -v "$PWD":/usr/local/apache2/htdocs/ httpd:latest`“.

3.4 Sieťový emulátor netem

Nástroj netem (network emulator) slúži na simulovanie rôznych parametrov siete, s ktorými sa môžeme stretnúť v rámci internetu, ako sú napr. oneskorenie alebo strata paketov. Tento nástroj je open-source, je súčasťou linuxového jadra od verzie 2.6 a nastavuje sa príkazom `tc`, ktorý je súčasťou linuxového balíčka iproute2 na konkrétnom rozhraní počítača.[22]

Parametre siete nastaviteľné nástrojom netem:

- **oneskorenie** rozsiahlych sietí (*WAN delay*) - umožňuje nám pridať oneskorenie spracovania paketov na konkrétnom sieťovom rozhraní vrátane variability oneskorenia s použitím normálneho rozdelenia pravdepodobnosti;
- **strata paketov** (*packet loss*) - nám, ako prezrádza samotný názov, dokáže náhodne zahadzovať pakety podľa nastavenej stratovosti;
- **duplikácia paketov** (*packet duplication*) - nastavuje mieru duplikácie paketov (podobne ako strata paketov);
- **chybovosť paketov** (*packet corruption*) - pridáva efekt šumu, čo má za následok vznik bitových chýb v paketoch;
- **zmena poradia paketov** (*packet reordering*) - prehadzuje poradie paketov v akom pôvodne prišli.

3.5 Program dig

Program dig (domain information groper) je nástroj príkazového riadku (terminálu) na dopytovanie sa na záznamy systému DNS, ktorý je vyvíjaný spoločne so serverom BIND a je dostupný v rámci balíku bind-utils, pričom sa dá použiť aj samostatne s ľubovoľným DNS serverom. Tento nástroj nám dokáže v porovnaní s nástrojom nslookup, ktorý už nie je aktívne vyvíjaný, zobraziť oveľa viac rozšírených informácií získaných z DNS záznamov. Podporuje taktiež filtrovanie na základe typu záznamu.[23]

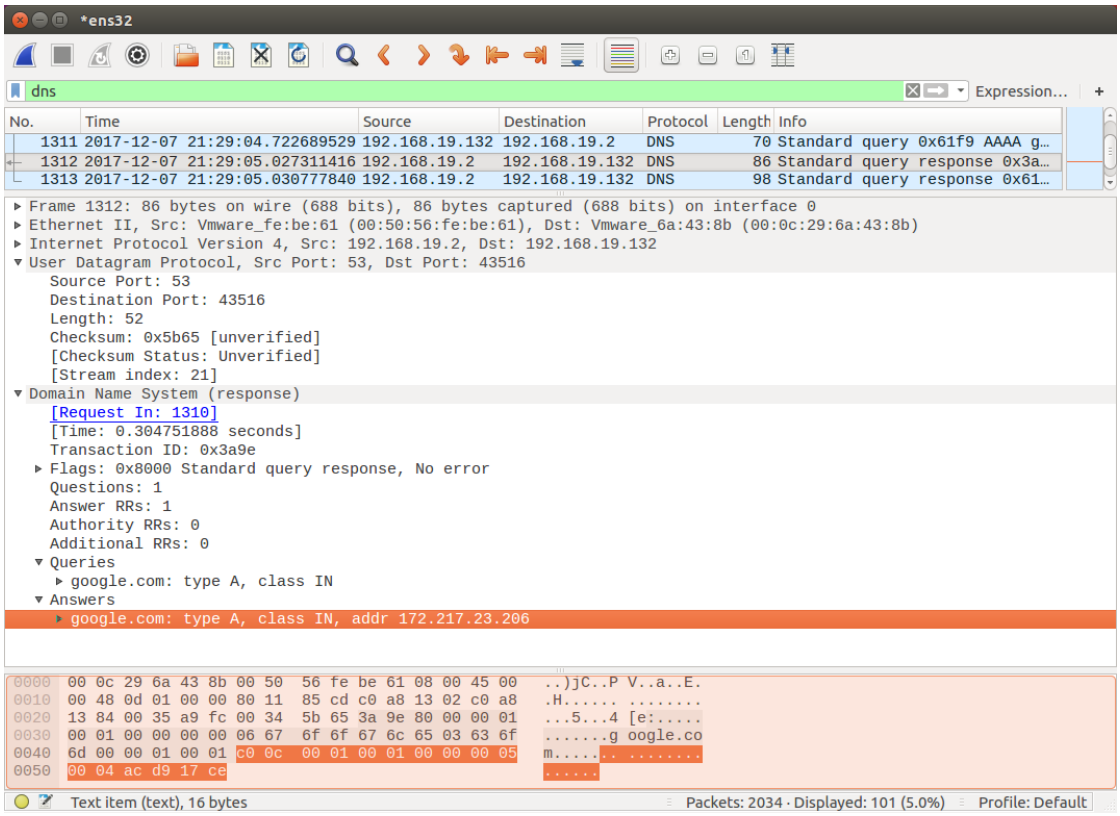
Spúšťa sa príkazom `dig` vrátane parametrov. Nápovedu získame príkazom „`dig -help`“. Napríklad pre získanie A záznamu (predvolené) domény `vutbr.cz` s použitím konkrétneho servera s IPv4 adresou `8.8.8.8` použijeme príkaz „`dig @8.8.8.8 vutbr.cz A`“. Po vykonaní tohto príkazu by sme mali dostať nasledujúci výstup:

Výpis 3.1: Výstup programu dig - dopyt na A záznamy

```
student@dns-lab:~$ dig @8.8.8.8 vutbr.cz A
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 20349
```

```
;; flags: qr rd ra ad; QUERY:1, ANSWER:1, AUTHORITY:0, ADDITIONAL:1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;vutbr.cz.       IN   A

;; ANSWER SECTION:
vutbr.cz.    299 IN   A 147.229.2.90

;; Query time: 74 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; MSG SIZE  rcvd: 53
```

3.6 Wireshark



Obr. 3.5: Aktívny filter a analýza paketu s DNS odpoveďou v programe Wireshark

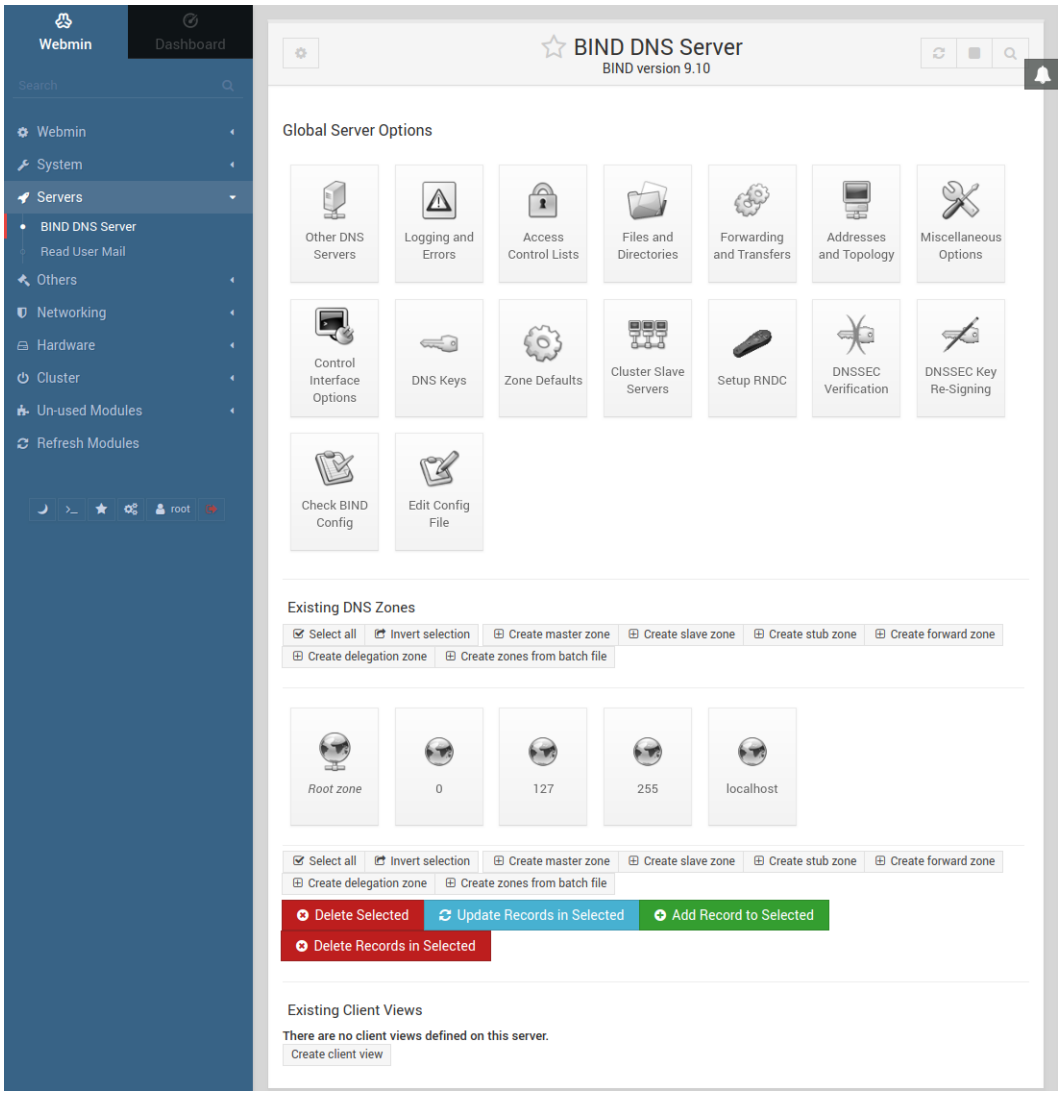
Wireshark je open-source analyzátor paketov vytvorený na hľadanie problémov a analýzu siete, vývoj softvéru a komunikačných protokolov, ako aj na edukačné účely. Dokáže zachytávať, okrem iného, komunikáciu na konkrétnom sieťovom rozhraní a neskôr túto komunikáciu analyzovať. Na zachytávanie paketov Wireshark používa knižnicu libpcap (Linux, BSD) alebo WinPcap (Windows). Táto knižnica do-

káže spolupracovať okrem Ethernet rozhrania aj s WLAN, Bluetooth, USB a mnoho ďalších.[24]

Wireshark podporuje taktiež filtrovanie paketov na základe rôznych kritérií, ako je napr. konkrétny protokol, sieťová adresa atď., pričom umožňuje využiť dva typy filtrov. DisplayFilter a Capture Filter. Prvý nám zo zachytených paketov zobrazí len tie, ktoré spĺňajú zadanú podmienku (zelený riadok na obrázku 3.5). Druhý sa nastavuje pred samotným zachytávaním, a na rozdiel od toho prvého, nám ovplyvňuje množstvo dát uložených do pamäte. Napr. v prípade, že potrebujeme analyzovať len pakety protokolu IPv4, nastavíme CaptureFilter. Neskôr si pri hľadaní konkrétneho paketu môžeme vypomôcť DisplayFiltrom, napr. podľa sieťovej adresy alebo konkrétneho protokolu.

3.7 Webové rozhranie Webmin

Webmin je webové rozhranie (obr. 3.6), ktoré uľahčuje správu rôznych služieb v OS Linux a Unix. Príkladom môže byť správa servera, v ktorom je nainštalovaný balík BIND. Okrem toho podporuje mnoho iných služieb a taktiež základné úkony nad operačným systémom.[25]

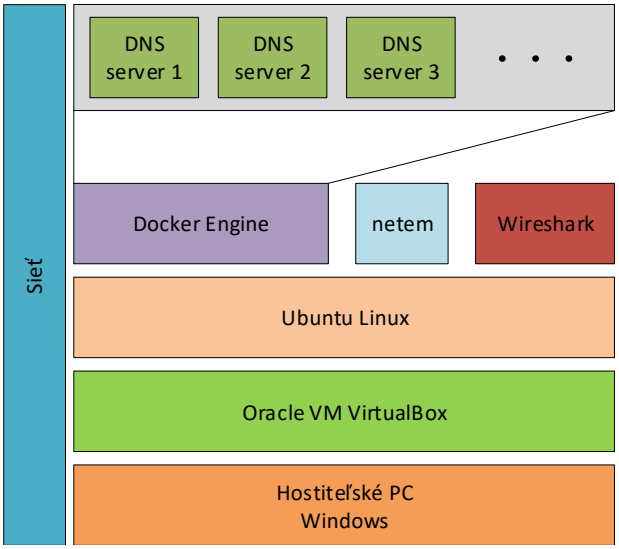


Obr. 3.6: Konfigurácia servera BIND vo webovom rozhraní Webmin

4 LABORATÓRNE ÚLOHY

4.1 Laboratórne prostredie

Laboratórne prostredie (obr. 4.1) je navrhnuté tak, aby študent, ktorý bude riešiť laboratórne úlohy, mal k dispozícii potrebné nástroje a aby sa tieto nástroje dali využiť čo najefektívnejšie na jednom počítači. Pre splnenie tohto účelu je použitá virtualizačná platforma Oracle VM VirtualBox vo verzii 5.2. Pomocou tejto platformy spúšťame virtuálny stroj s nainštalovaným OS Ubuntu Desktop 16.04.4 LTS. Verziu s dlhodobou podporou sme zvolili z dôvodu lepšej kompatibility s niektorými verziami softvéru. Táto verzia je podporovaná do apríla 2021[18]. V tomto virtuálnom OS máme nainštalovaný Docker v aktuálnej verzii 18.03.1-ce (komunitná verzia) v ktorom vytvárame jednotlivé kontajnery určené na spúšťanie nezávislých inštancií DNS serverov (BIND). Taktiež je dostupný program Wireshark 2.4.6 vrátane libpcap 1.7.4. Program dig je súčasťou základnej inštalácie Ubuntu Desktop, ako aj nástroj netem, ktorý je súčasťou linuxového jadra.



Obr. 4.1: Fyzická topológia laboratórnej siete

Ako DNS server bol zvolený BIND a je použitý viacnásobne pomocou kontajnerov v Dockeri. Ako image sme využili trochu upravený obraz sameersbn/bind[26] dostupný na internete (Docker Hub). Tento obraz má okrem samotného servera BIND v sebe zakomponovaný aj Webmin, čo nám umožňuje upravovať konfiguráciu servera dvomi spôsobmi (webové rozhranie a konfiguračné súbory). Úpravy sa týkali zmeny prístupných konfiguračných priečinkov jednotlivých kontajnerov a tiež inštalácia balíka iproute (netem) a iputils-ping.

4.2 Virtuálny systém Ubuntu Desktop

V nainštalovanom OS Ubuntu sú vytvorení 2 používatelia:

- `spravca` (heslo `aaa`) - správca systému (sudoer),
- `student` (heslo `student`) - bežný používateľ.

Príkaz `docker` môže spúšťať aj používateľ `student`. Skripty sa nachádzajú v priečinku `/scripts/`. Predpripravené konfiguračné súbory pre DNS servery sa nachádzajú v priečinku `/dns-lab-defaults/`. Priečinok `/dns-lab/` uchováva konfiguráciu práve spustených serverov (na priamu úpravu). Niektoré skripty a konfiguračné súbory je možné nájsť v prílohe C.

Systém ma pripojené 2 virtuálne rozhrania. Prvé je typu NAT pre pripojenie na internet (interne `enp0s3`). Druhé je typu Host-only a využíva rozsahy adries `10.0.0.0/24` a `fd00::/64` (interne `enp0s8`). Po štarte systému sa spúšťa skript `dns_lab-set_network.sh`, ktorý nastavuje IP adresy vo virtuálnom systéme na rozhraní `macvlan0@enp0s8` (výpis v prílohe C.1). Na túto sieť sa podobne pripájajú kontajnery s definovanou sieťou `macvlan-net`.

Obraz systému bol vytvorený na virtualizačnej platforme Oracle VM VirtualBox. Počíta sa s tým, že virtuálny stroj nebude uchovávať zmeny po vypnutí. Obraz bude k dispozícii na stiahnutie v službe CESNET ownCloud.

4.3 Navrhnuté laboratórne úlohy

Laboratórne úlohy vrátane popisu logickej topológie, postupov a vzorového riešenia sú v prílohách A a B.

4.3.1 Ciele 1. laboratórnej úlohy

Cieľom tejto úlohy je študentovi experimentálne poukázať na niektoré základné vlastnosti systému DNS a to na vlastnej DNS hierarchii. Hlavné ciele úlohy:

- zoznámenie sa so systémom DNS a serverom BIND,
- objasnenie funkcie normálneho a reverzného prekladu v DNS,
- objasnenie princípu rekurzie v DNS a ukladanie do cache,
- vytvorenie novej zóny a pridanie záznamov,
- nastavenie sekundárneho doménového servera.

4.3.2 Ciele 2. laboratórnej úlohy

Cieľom tejto úlohy je v prvom rade ozrejmiť študentovi funkciu DNSSEC-u v systéme DNS a tiež poukázať na súvisiace problémy. Hlavné ciele úlohy:

- zoznámenie sa s DNSSEC,

- nastavenie rekurzívneho resolvera,
- overenie DNSSEC validácie,
- reťazec dôvery v DNSSEC,
- sledovanie správania transportnej vrstvy,
- porovnanie výkonu verejných DNS serverov.

5 ZÁVER

Cieľom tejto práce bolo zoznámiť sa s fungovaním systému DNS (*Domain Name System*) a s jeho možnosťami využitia v rámci internetu. Bola popísaná základná štruktúra systému, zloženie doménových názvov, spôsob ukladania doménových informácií v databáze DNS a v neposlednom rade hierarchia a typy serverov DNS. Ďalej bol popísaný protokol DNS a základné typy správ, ktoré tento protokol typu klient-server využíva. V ďalšej časti sú vymenované a stručne popísané základné vety *Resource Record*. Ďalej je naznačený postup prekladu domén vrátane reverzného prekladu. Popísane je taktiež bezpečnostné rozšírenie DNSSEC, ktoré pomáha udržať systém DNS ako doveryhodný.

Práca sa taktiež venuje softvérovým implementáciám DNS serverov, kam patria rekurzívne aj autoritatívne servery, ako sú napríklad najviac používaný BIND alebo Knot DNS, ktorý je vyvíjaný českým združením *CZ.NIC*. Takisto boli popísané niektoré nástroje a technológie, ktoré sú využité pri návrhu laboratórnych úloh a vytváraní vhodného laboratórneho prostredia.

V závere práce je popísané laboratórne prostredie vrátane pripraveného obrazu operačného systému, ktorý je predkonfigurovaný na použitie v laboratóriu. Podarilo sa vytvoriť veľmi univerzálny systém, ktorý je možné rozširovať pre rôzne ďalšie úlohy. Využitá bola ako virtualizácia tak aj kontajnery. Súčasťou práce bol návrh a realizáciách dvoch laboratórnych úloh, vrátane popisu topológie, postupu práce a vzorového riešenia. Bol kladený dôraz na dostatočné ozrejmenie základných princípov systému DNS.

LITERATÚRA

[1] KABELOVÁ, Alena a Libor DOSTÁLEK. *Velký průvodce protokoly TCP/IP a systémem DNS.* 5., aktualiz. vyd. Brno: Computer Press, 2008. ISBN 978-80-251-2236-5.

[2] JEŘÁBEK, Jan. *Pokročilé komunikační techniky.* Verze 2017. Brno: Vysoké učení technické v Brně, 2015. ISBN 978-80-214-4713-4.

[3] IDN ccTLD Fast Track String Evaluation Completion. *Internet Corporation for Assigned Names and Numbers* [online]. Posledná úprava 19. 2. 2014 [cit. 9. 12. 2017]. Dostupné z URL: <`https://www.icann.org/resources/pages/string-evaluation-completion-2014-02-19-en`>.

[4] .ARPA Zone Management. *Internet Assigned Numbers Authority* [online]. [cit. 9. 12. 2017]. Dostupné z URL: <`https://www.iana.org/domains/arpa`>.

[5] Special-Use Domain Names. *Internet Assigned Numbers Authority* [online]. Posledná úprava 27. 10. 2015 [cit. 9. 12. 2017]. Dostupné z URL: <`https://www.iana.org/assignments/special-use-domain-names/special-use-domain-names.xhtml`>.

[6] About the Program. *ICANN New gTLDs* [online]. [cit. 9. 12. 2017]. Dostupné z URL: <`https://newgtlds.icann.org/en/about/program`>.

[7] IDN - Internationalized domain names. *CZ.NIC* [online]. [cit. 9. 12. 2017]. Dostupné z URL: <`https://www.xn--hkyrky-ptac70bc.cz`>.

[8] Jak funguje DNSSEC. *CZ.NIC* [online]. [cit. 15. 5. 2018]. Dostupné z URL: <`https://www.dnssec.cz/page/444/jak-funguje-dnssec`>.

[9] Trust Anchors and Keys. *Internet Assigned Numbers Authority* [online]. [cit. 15. 5. 2018]. Dostupné z URL: <`https://www.iana.org/dnssec/files`>.

[10] DNSSEC – What Is It and Why Is It Important? *Internet Corporation for Assigned Names and Numbers* [online]. Posledná úprava 29. 1. 2014 [cit. 15. 5. 2018]. Dostupné z URL: <`https://www.icann.org/resources/pages/dnssec-qaa-2014-01-29-en`>.

[11] BIND Open Source DNS Server. *Internet Systems Consortium* [online]. Posledná úprava 28. 7. 2017 [cit. 12. 12. 2017]. Dostupné z URL: <`https://www.isc.org/downloads/bind`>.

[12] Name Server Daemon (NSD). *NLnet Labs* [online]. Posledná úprava 11. 12. 2017 [cit. 12. 12. 2017]. Dostupné z URL: <https://www.nlnetlabs.nl/projects/nsd>.

[13] New Architecture Model for K-root Local Instances. *RIPE Labs* [online]. Posledná úprava 11. 3. 2015 [cit. 12. 12. 2017]. Dostupné z URL: <https://labs.ripe.net/Members/romeo_zwart/new-architecture-for-k-root-local-nodes>.

[14] L-Root. *ICANN DNS Engineering* [online]. [cit. 12. 12. 2017]. Dostupné z URL: <https://www.dns.icann.org/lroot/>.

[15] Knot DNS (software). *CZ.NIC* [online]. Posledná úprava 24. 11. 2017 [cit. 12. 12. 2017]. Dostupné z URL: <https://www.knot-dns.cz/>.

[16] About Unbound. *Unbound* [online]. Posledná úprava 10. 10. 2017 [cit. 12. 12. 2017]. Dostupné z URL: <https://www.unbound.net/>.

[17] Knot Resolver (software). *CZ.NIC* [online]. Posledná úprava 12. 12. 2017 [cit. 12. 12. 2017]. Dostupné z URL: <https://www.knot-resolver.cz/>.

[18] Home. *Ubuntu Wiki* [online]. Posledná úprava 9. 5. 2018 [cit. 11. 5. 2018]. Dostupné z URL: <https://wiki.ubuntu.com>.

[19] BionicBeaver/ReleaseNotes. *Ubuntu Wiki* [online]. Posledná úprava 4. 5. 2018 [cit. 11. 5. 2018]. Dostupné z URL: <https://wiki.ubuntu.com/BionicBeaver/ReleaseNotes>.

[20] Oracle VM VirtualBox. *Documentation* [online]. Posledná úprava 07. 8. 2017 [cit. 21. 4. 2018]. Dostupné z URL: <https://www.virtualbox.org/wiki/Documentation>.

[21] Get Started, Part 1: Orientation and setup. *Docker Documentation* [online]. Posledná úprava 10. 11. 2017 [cit. 8. 12. 2017]. Dostupné z URL: <https://docs.docker.com/get-started/>.

[22] Networking:netem. *Linux Foundation Wiki* [online]. Posledná úprava 19. 7. 2016 [cit. 4. 12. 2017]. Dostupné z URL: <https://wiki.linuxfoundation.org/networking/netem>.

[23] dig - Manual pages. *Internet Systems Consortium* [online]. Posledná úprava 28. 7. 2017 [cit. 8. 12. 2017]. Dostupné z URL: <ftp://ftp.isc.org/isc/bind9/cur/9.10/doc/arm/man.dig.html>.

[24] Documentation. *Wireshark* [online]. Posledná úprava 9. 11. 2014 [cit. 8. 12. 2017]. Dostupné z URL: <https://www.wireshark.org/docs/>.

[25] Introduction. *Webmin Documentation* [online]. Posledná úprava 28. 10. 2017 [cit. 17. 5. 2017]. Dostupné z URL: <https://doxfer.webmin.com/Webmin/Main_Page>.

[26] docker-bind/README.md at master. *sameersbn/docker-bind* [online]. Posledná úprava 26. 6. 2017 [cit. 12. 12. 2017]. Dostupné z URL: <https://github.com/sameersbn/docker-bind/blob/master/README.md#introduction>.

[27] IPv6 with Docker. *Docker Documentation* [online]. Posledná úprava 20. 11. 2017 [cit. 12. 12. 2017]. Dostupné z URL: <https://docs.docker.com/engine/userguide/networking/default_network/ipv6/>.

ZOZNAM SYMBOLOV, VELIČÍN A SKRATIEK

AXFR	Asynchronous Transfer Full Range
BIND	Berkeley Internet Name Domain
dig	domain information groper
DNS	Domain Name System
DNSSEC	Domain Name System Security Extensions
FQDN	Fully Qualified Domain Name
GUA	Global Unicast Address
IDN	Internationalized Domain Names
IP	Internet Protocol
IPv6	Internet Protocol version 6
IXFR	Incremental Zone Transfer
NSD	Name Server Daemon
RR	Resource Record
TCP	Transport Control Protocol
UDP	User Datagram Protocol
ULA	Unique Local Address
VM	Virtual Machine

ZOZNAM PRÍLOH

A LABORATÓRNA ÚLOHA 1

A.1 Návod

A.1.1 Ciele laboratórnej úlohy

- zoznámenie sa so systémom DNS a serverom BIND,
- objasnenie funkcie normálneho a reverzného prekladu v DNS,
- objasnenie princípu rekurzie v DNS a ukladanie do cache,
- vytvorenie novej zóny a pridanie záznamov,
- nastavenie sekundárneho doménového servera.

A.1.2 Teoretický úvod

Systém DNS

Systém DNS nám umožňuje používať tzv. doménové názvy namiesto IP adries všade, kde máme prístup k databáze DNS. Doménou označujeme skupinu počítačov, ktoré sú v určitom vzťahu (napr. tvoria organizačnú jednotku alebo zdieľajú geografickú polohu). Táto databáza je decentralizovaná, má stromovú štruktúru a informácie o jednotlivých doménach poskytujú doménové servery.[1, s. 255]

Informácie o doménových menách, či už ide o príslušné IP adresy, ale aj všetky ostatné informácie, sú uložené v pamäti DNS serverov v tvare zdrojových viet (*Resource Records* - **RR**). Nasledujúci zoznam stručne popisuje základné a najčastejšie používané typy viet.[1, 2]

- **SOA** (*Start of Authority*) - záznam je práve jeden vždy na začiatku konfiguračného súboru zóny a určuje menný server, ktorý je autoritatívnym zdrojom informácii pre danú doménu.
- **A** (*A host address*) - záznam poskytuje mapovanie doménového mena na 32 bitovú IPv4 adresu.
- **AAAA** (*IP6 address*) - podobne ako A záznam, ale používa 128 bitové IPv6 adresy.
- **MX** (*Mail eXchange*) - obsahuje dve informácie: 16 bitové číslo ako preferenciu a doménové meno e-mailového servera.
- **CNAME** (*Canonical name for an alias*) - doménové meno, ktoré špecifikuje alias (synonymum) k inému menu. Môže odkazovať iba na A/AAAA záznam, nie iný CNAME.
- **NS** (*Authoritative name server*) - definuje autoritatívne doménové servery pre danú doménu. Musí odkazovať na doménové meno, ktorému je priradená IP adresa pomocou A/AAAA záznamu (podobne ako pri CNAME).

- **PTR** (*Domain name pointer*) - slúži k prekladu IP adresy na doménové meno (reverzný preklad), tj. k prekladu prvkov domény `in-addr.arpa` (IPv4) prípadne `ip6.arpa` (IPv6).
- **TXT** (*Text string*) - obsahuje vo svojej dátovej časti všeobecný textový reťazec.
- **SRV** (*Service locator*) - má za úlohu poskytovať k názvom počítačov taktiež názvy služieb na konkrétnej doméne. Tieto záznamy využívajú napr. systémy Microsoftu (LDAP).

BIND

Medzi najznámejšie a najviac používané softvérové implementácie DNS protokolu patrí jednoznačne **BIND** (*Berkeley Internet Name Domain*), ktorý vznikol v osemdesiatych rokoch na Kalifornskej univerzite v Berkeley a v súčasnosti je vyvíjaný konzorciom **ISC** (*Internet Software Consortium*). BIND, niekedy označovaný ako `named`, je dostupný v podobe zdrojového kódu pre Unix-like operačné systémy, kde sa BIND považuje za de facto štandard, a takisto pre Windows v podobe spustiteľných súborov. BIND je najviac používaný autoritatívny server v koreňovej zóne. Posledná verzia 9 bola vytvorená už v roku 2000, aktuálne to je 9.11.2.[11]

BIND je open-source softvérový balík zložený z týchto častí:

- doménový resolver,
- autoritatívny DNS server,
- nástroje ako napr. `dig`, `nslookup`, `host` atď.

Resolver sa stará o získavanie informácií od vzdialených autoritatívnych serverov. Tento resolver (použitý ako rekurzívny server) potom preposiela odpoveď stub resolverom, ktoré sú súčasťou OS. Knižnica BIND resolveru býva taktiež často použitá ako stub resolver v mnohých distribúciach OS Linux. **Autoritatívny** DNS server naopak odpovedá na žiadosti od resolverov a poskytuje im informácie, ktoré sú pre daný server autoritatívne. **Nástroje** slúžia administrátorom na uľahčenie správy DNS serverov, použiteľné všeobecne v systéme DNS.

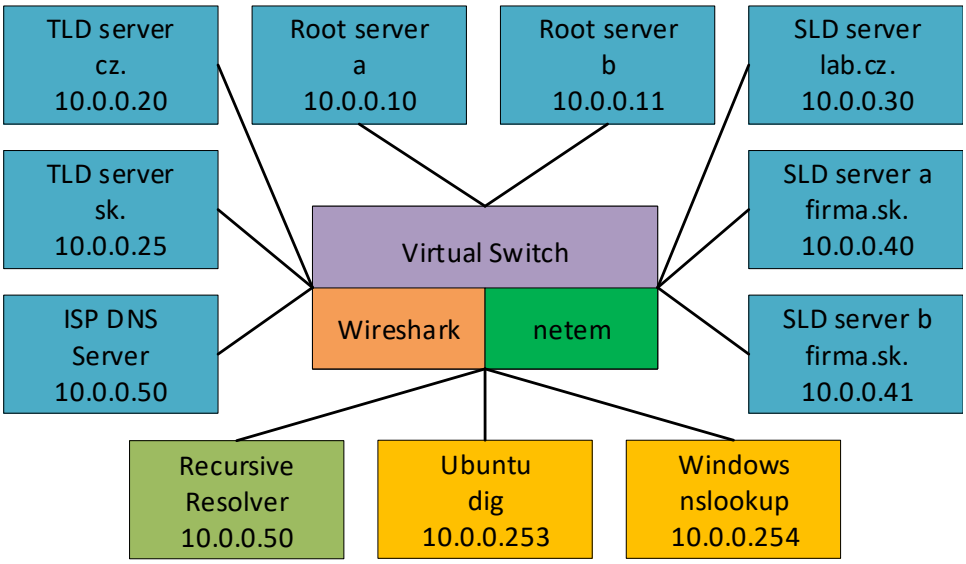
BIND ukladá dáta a nastavenia v týchto umiestneniach:

- `/etc/bind/` - konfiguračné súbory, ktoré sa dlhodobo nemenia,
- `/var/lib/bind/` - zónové súbory primárnych zón,
- `/var/cache/bind/` - zónové súbory sekundárnych zón, prípadne automaticky podpisované primárne zóny (DNSSEC),
- pamäť RAM - vyrovnávacia pamať (cache).

A.1.3 Laboratórne prostredie

Laboratórna úloha je pripravená v prostredí operačného systému Ubuntu Desktop 16.04 virtualizovaného na platforme Oracle VM VirtualBox. Prihlasovacie údaje do systému sú `student:student`. Zmeny vykonané vo virtualizovanom OS sú uchovávané len do reštartu virtuálneho stroja. Tento systém obsahuje program dig (terminál), Wireshark (GUI) a niekoľko predkonfigurovaných inštancií DNS serverov (BIND). Z hostiteľského OS (Windows) je možné využiť nástroj príkazového riadku nslookup. Na pracovnej ploche Ubuntu nájdete odkazy na tri skripty. `Start LAB1`, `Start LAB2` na spustenie jednotlivých laboratórnych úloh a `Restart LAB1` na reštartovanie DNS serverov 1. laboratórnej úlohy so zachovaním konfigurácie. Skript spustíte dvojklikom a následne vyberiete možnosť *Run in Terminal*. Prístup na webové rozhranie Webmin je možný pomocou adresy v tvare `https://<ip_adresa>:10000` alebo pomocou odkazov na ploche. Hlásenie o nedôveryhodnom (self signed) certifikáte ignorujeme. Prihlasovacie údaje do Webminu sú `root:password`. Lokálne IP adresy pre virtualizované Ubuntu sú `10.0.0.253, fd00::253`, pre hostiteľský OS Windows to sú `10.0.0.254, fd00::254`. Okrem webového rozhrania je možné upravovať konfiguračné súbory jednotlivých inštancií BIND-u. Tie sa nachádzajú v umiestnení `/dns-lab/<nazov_servera>/bind` (odkaz na ploche). V tomto priečinku sú podpriečinky `etc` pre `/etc/bind`, `lib` pre `/var/lib/bind` a `cache` pre `/var/cache/bind`.

A.1.4 Počiatočná konfigurácia



Obr. A.1: Logická topológia laboratórnej siete

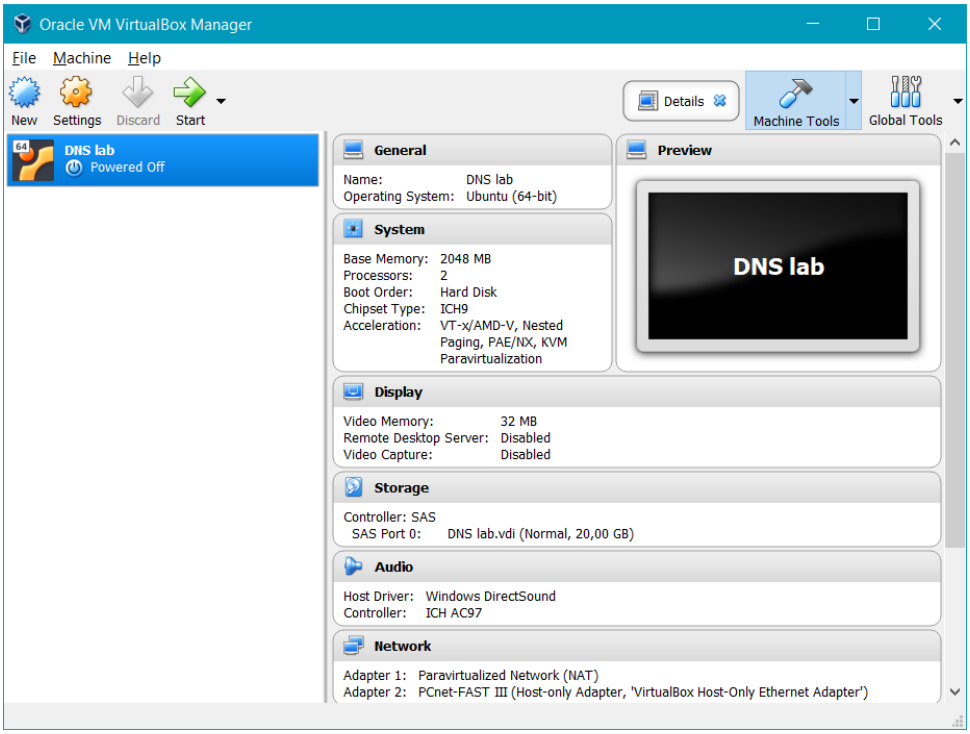
Tab. A.1: DNS servery a ich IP adresy

Názov servera	**IPv4**	**IPv6**	**Obsahuje zóny**
`server-root-a`	10.0.0.10	fd00::10	.(root), arpa, in-addr.arpa, ip6.arpa
`server-root-b`	10.0.0.11	fd00::11	.(root), arpa, in-addr.arpa, ip6.arpa
`server-cz`	10.0.0.20	fd00::20	cz
`server-sk`	10.0.0.25	fd00::25	sk
`server-lab_cz`	10.0.0.30	fd00::30	lab.cz
`server-firma_sk-a`	10.0.0.40	fd00::40	firma.sk, 18.200.199.in-addr.arpa, b.a.0.9.8.7.6.5.4.3.2.1.1.0.0.2.ip6.arpa
`server-firma_sk-b`	10.0.0.41	fd00::41	
`server-isp`	10.0.0.50	fd00::50	200.199.in-addr-arpa, 8.7.6.5.4.3.2.1.1.0.0.2.ip6.arpa
`recursive-resolver`	10.0.0.100	fd00::100	

Logická topológia je znázornená na obrázku A.1, kde je možné vidieť niekoľko autoritatívnych serverov, rekurzívny resolver (s cache) aj samotných klientov a využívané nástroje. Všetky doménové servery, s výnimkou rekurzívneho resolvera, majú nastavené oneskorenie siete pomocou nástroja netem na 10 ms (simulácia bežného internetu). Každá inštancia DNS servera má vlastnú IPv4 a IPv6 adresu (viď tab. A.1).

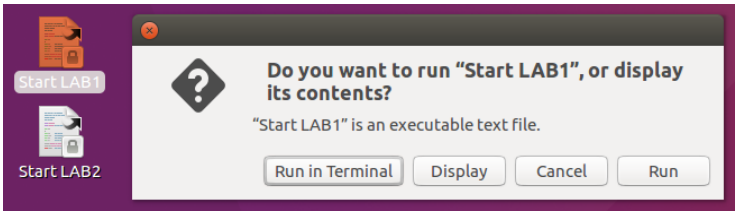
Na začiatku tejto úlohy je funkčná základná DNS hierarchia vrátane koreňovej zóny, dvoch TLD domén (cz. a sk.) a v rámci nich dve domény druhej úrovne (lab.cz a firma.sk). Takisto je funkčný reverzný preklad na niekoľkých IP adresách (in-addr.arpa. a ip6.arpa.). Patria tu rozsahy `199.200.0.0/16`, `199.200.18.0/24`, `2001:1234:5678::/48`, `2001:1234:5678:90ab::/64`. Tieto sú delegované na `server-isp` a `server-firma-a`. DNSSEC validácia je vypnutá. Rekurzívny resolver je plne funkčný a ukladá záznamy do cache.

A.1.5 Postup

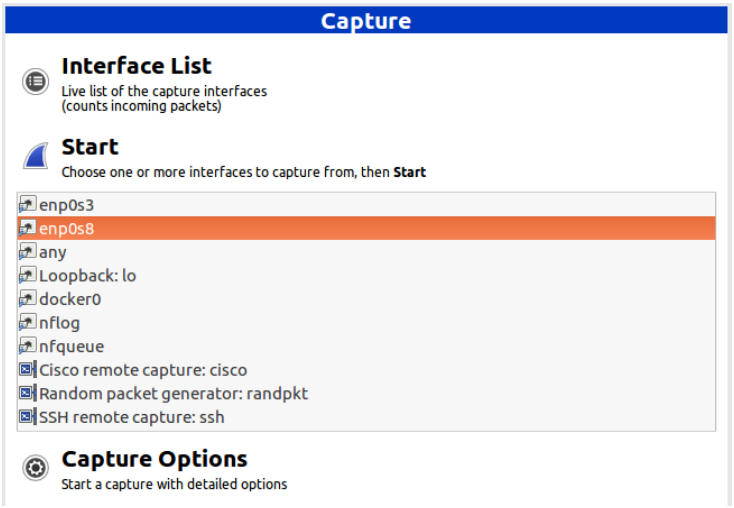


Obr. A.2: Základné okno programu Oracle VM VirtualBox

Spustíme program Oracle VM VirtualBox a v hlavnom okne vyberieme a spustíme virtuálny stroj (VM) s názvom *DNS lab* (obr. A.2). Po prihlásení pod účtom `student` (heslo `student`) spustíme skript `Start LAB1` nachádzajúci sa na pracovnej ploche (obr. A.3) a počkáme, kým nezmizne terminálové okno. Pre analyzovanie DNS spustíme vo virtuálnom systéme Wireshark a aktivujeme odchytávanie paketov na rozhraní `enp0s8` (obr. A.4). Potom aktivujeme filter „`dns`“ a potvrdíme tlačidlom *Apply* (obr. A.5).



Obr. A.3: Spustenie skriptu `Start Lab1`



Obr. A.4: Výber rozhrania na zachytávanie paketov vo Wiresharku

Obr. A.5: Nastavenie filtra vo Wiresharku pre zobrazenie DNS správ

Základný preklad

V úvodnej časti si overíme základnú funkciu systému DNS a v programe Wireshark si zobrazíme správy, ktoré medzi sebou posielajú klient a doménové servery. Uistíme sa, že máme spustené zachytávanie paketov vo Wiresharku. Na získavanie informácií z doménových serverov použijeme program dig. Vo virtuálnom systéme otvoríme terminál a spustíme príkaz „`dig -help`“, ktorý nám vypíše príručku k použitiu programu dig. Keď sme si príručku dostatočne prezreli, spustíme príkaz:

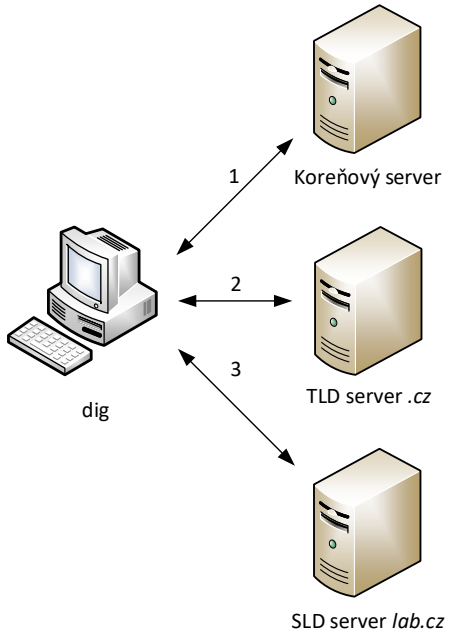
„`dig @10.0.0.10 test.lab.cz.`“.

Pomocou tohto príkazu sa pýtame doménového servera so zadanou IP adresou (v tomto prípade koreňový server) na A záznam zadaného doménového mena (`test.lab.cz`). Ak nedefinujeme konkrétny typ záznamu pomocou parametru, program dig predpokladá A záznam. Bodkou na konci tohto mena určujeme, že to je úplné doménové meno. Ako vidíme vo výsledku (výpis. A.1), koreňový server nám neodpovedal A záznamom, ale odkázal nás na doménový server `a.ns.nic.cz` a pripojil k odpovedi aj IP adresy daného servera. Upozornenie o nedostupnej rekurzii je vo výstupe preto, že program dig ju štandardne využíva, no na tomto serveri je zakázaná. Tento stav je na autoritatívnych serveroch bežný. Ak by mali tieto verejné servery rekurziu povolenú, boli by náchylné na amplifikačné útoky. Toto

prednastavené správanie programu dig ošetríme v nasledujúcich dopytoch parametrom `+norecurse`. Vo Wiresharku vidíme jednoduchú komunikáciu otázka/odpoveď (obr. A.6).

```
10.0.0.254 10.0.0.10     DNS          82 Standard query 0xe9ac A test.lab.cz OPT
10.0.0.10  10.0.0.254    DNS         149 Standard query response 0xe9ac A test.lab.cz NS a.ns.nic.cz A 10.0.0.20 AAAA fd00::20 OPT
```

Obr. A.6: Zachytené správy vo Wiresharku



Obr. A.7: Iteratívne dopyty

Teraz príkaz zopakujeme, no spýtame sa doménového servera, ktorý nám bol ponúknutý v predchádzajúcej odpovedi (obr. A.7. Môžeme použiť aj IPv6 adresy. Postupne by sme sa takto mali krok za krokom dostať k hľadanému A záznamu. Dopytovať sa teda budeme iteratívne. Zatiaľ nevyužívame rekurzívny resolver.

Výpis A.1: Výstup programu dig - dopyt na A záznamy

```
student@dns-lab:~$ dig @10.0.0.10 test.lab.cz.
; <<>> DiG 9.10.3-P4-Ubuntu <<>> @10.0.0.10 test.lab.cz.
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 44785
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 3
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;test.lab.cz.      IN   A
```

```
;; AUTHORITY SECTION:
cz.     300 IN  NS  a.ns.nic.cz.

;; ADDITIONAL SECTION:
a.ns.nic.cz.    300 IN  A 10.0.0.20
a.ns.nic.cz.    300 IN  AAAA  fd00::20

;; Query time: 20 msec
;; SERVER: 10.0.0.10#53(10.0.0.10)
;; MSG SIZE  rcvd: 107
```

Rovnaký postup si môžeme vyskúšať aj s nástrojom nslookup v hostiteľskom OS Windows s využitím IPv6 adresy druhého koreňového servera týmto príkazom:

„`nslookup -norecurse test.lab.cz. fd00::11`“,

ktorý nám vráti odpoveď znázornenú vo výpise A.2.

Výpis A.2: Výstup programu nslookup - dopyt na A/AAAA záznamy

```
C:\>nslookup -norecurse test.lab.cz. fd00::11
Server:  UnKnown
Address:  fd00::11

Name:    test.lab.cz
Served by:
- a.ns.nic.cz
          10.0.0.20
          fd00::20
          cz
```

Na záver tejto časti si vo Wiresharku vyberieme jeden DNS dopyt a k nemu odpoveď. Všimneme si, čo všetko sa prenáša v týchto DNS správach.

Otázka 1: Akú IP adresu sme zistili pre doménové meno `test.lab.cz`? Existuje pre toto meno aj AAAA záznam? Ak áno, akú adresu obsahuje?
Otázka 2: Prečo nám neodpovedal A záznamom priamo koreňový server?
Otázka 3: Aký transportný protokol a port bol využitý?
Otázka 4: Akú veľkosť mali prenesené správy po odrátaní réžie pri prvom dopyte (otázka/odpoveď)?

Reverzný preklad

V tejto časti si predvedieme reverzný preklad, tzn. preklad IP adries na doménové meno. Budeme postupovať podobným spôsobom ako v predchádzajúcej časti. Reverzné záznamy pre IPv4 adresy nájdeme v doméne `in-addr.arpa`, pre IPv6 v doméne `ip6.arpa`. Tentokrát budeme hľadať reverzný záznam (PTR) pre IP adresu `199.200.18.1`. Spustíme príkaz:

„`dig @10.0.0.10 -x 199.200.18.1 +norecurse`“.

Tento príkaz je skrátená verzia nasledujúceho príkazu:

„`dig @10.0.0.10 1.18.200.199.in-addr.arpa. PTR +norecurse`“.

V odpovedi vidíme (výpis A.3), že nás server odkázal na iný server `ns1.isp.sk`, neposlal nám však jeho IP adresu. Tú si musíme zistiť:

„`dig @10.0.0.10 ns1.isp.sk. +norecurse`“.

V tomto prípade stále nevieme požadovanú adresu, musíme sa spýtať navrhovaného servera:

„`dig @10.0.0.25 ns1.isp.sk. +norecurse`“.

Tento nám vrátil adresu `10.0.0.50`. Na tejto adrese budeme pokračovať v hľadaní nášho reverzného záznamu. Zopakujeme tento postup, až kým nezískame PTR záznam pre IP adresu `199.200.18.1`.

Výpis A.3: Výstup programu dig - dopyt na PTR záznam

```
student@dns-lab:~$ dig @10.0.0.10 -x 199.200.18.1 +norecurse
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 28323
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;1.18.200.199.in-addr.arpa. IN  PTR

;; AUTHORITY SECTION:
200.199.in-addr.arpa. 300 IN  NS  ns1.isp.sk.

;; Query time: 10 msec
;; SERVER: 10.0.0.10#53(10.0.0.10)
;; MSG SIZE  rcvd: 78
```

Na záver tejto časti, podobne ako v tej predtým, si vo Wiresharku vyberieme jeden DNS dopyt a k nemu odpoveď. Všimneme si, čo všetko sa prenáša v týchto DNS správach a porovnáme s minulými správami.

Otázka 5: Aké bolo zistené doménové meno pre IP adresu `199.200.18.1`?

Rekurzívny resolver a využitie cache pamäte

V prvých dvoch častiach úlohy sme prechádzali distribuovanú databázu DNS manuálne a v niektorých prípadoch to zabralo značné množstvo času a úsilia. Aby sme to nemuseli robiť takto manuálne, existujú rekurzívne resolvery, ktoré sú klasické DNS

servery ale nakonfigurované tak, aby práve takýmto spôsobom prechádzali túto databázu a odpovedali na otázku priamo s výsledkom, ktorý sme požadovali. Takéto servery využívaju väčšinou poskytovatelia internetových služieb.

Ak spustíme ten istý príkaz ako v prvej časti úlohy s tým, že dopyt nasmerujeme na rekurzívny resolver:

„`dig @10.0.0.100 test.lab.cz.`“,

dostaneme ihneď očakávanú odpoveď. Rekurzívny resolver vyhľadá záznam obdobným spôsobom, ako sme to urobili predtým manuálne a vráti finálnu odpoveď, pričom si získanú informáciu uloží do pamäte cache. V prípade, že údaje sa nachádzajú v cache, pošle odpoveď ešte v kratšom čase. Vo Wiresharku si môžeme prezrieť (obr. A.8), ako rekurzívny resolver postupoval a ako komunikoval s ostatnými autoritatívnymi doménovými servermi. Rovnako to môžeme vyskúšať na reverznom zázname, ktorý sme hľadali predtým:

„`dig @10.0.0.100 -x 199.200.18.1`“.

Záznamy majú predvolené TTL nastavené na 5 minút. Keď je potrebné cache vymazať, dobre poslúži nasledujúci príkaz:

„`docker exec -it dns-lab.recursive-resolver rndc flush`“.

```
10.0.0.253 10.0.0.100    DNS          96 Standard query 0xd662 PTR 1.18.200.199.in-addr.arpa OPT
10.0.0.100 10.0.0.11     DNS          96 Standard query 0xd1cb PTR 1.18.200.199.in-addr.arpa OPT
10.0.0.100 10.0.0.11     DNS          70 Standard query 0x5c5f NS <Root> OPT
10.0.0.11  10.0.0.100    DNS         120 Standard query response 0xd1cb PTR 1.18.200.199.in-addr.arpa NS ns1.isp.sk OPT
10.0.0.11  10.0.0.100    DNS         204 Standard query response 0x5c5f NS <Root> NS a.root-servers.net NS b.root-servers.net A 10.0.0.10
10.0.0.100 10.0.0.10     DNS          81 Standard query 0x61a2 A ns1.isp.sk OPT
10.0.0.100 10.0.0.10     DNS          81 Standard query 0x4fe0 AAAA ns1.isp.sk OPT
10.0.0.10  10.0.0.100    DNS         145 Standard query response 0x61a2 A ns1.isp.sk NS a.tld.sk A 10.0.0.25 AAAA fd00::25 OPT
10.0.0.10  10.0.0.100    DNS         145 Standard query response 0x4fe0 AAAA ns1.isp.sk NS a.tld.sk A 10.0.0.25 AAAA fd00::25 OPT
10.0.0.100 10.0.0.25     DNS          81 Standard query 0x36f9 A ns1.isp.sk OPT
10.0.0.100 10.0.0.25     DNS          81 Standard query 0x630b AAAA ns1.isp.sk OPT
10.0.0.25  10.0.0.100    DNS          97 Standard query response 0x36f9 A ns1.isp.sk A 10.0.0.50 OPT
10.0.0.25  10.0.0.100    DNS         109 Standard query response 0x630b AAAA ns1.isp.sk AAAA fd00::50 OPT
10.0.0.100 10.0.0.50     DNS          96 Standard query 0x951a PTR 1.18.200.199.in-addr.arpa OPT
10.0.0.50  10.0.0.100    DNS         122 Standard query response 0x951a PTR 1.18.200.199.in-addr.arpa NS ns1.firma.sk OPT
fd00::100  fd00::25      DNS         103 Standard query 0x154d A ns1.firma.sk OPT
fd00::100  fd00::25      DNS         103 Standard query 0x0489 AAAA ns1.firma.sk OPT
fd00::25   fd00::100     DNS         161 Standard query response 0x154d A ns1.firma.sk NS ns1.firma.sk A 10.0.0.40 AAAA fd00::40 OPT
fd00::25   fd00::100     DNS         161 Standard query response 0x0489 AAAA ns1.firma.sk NS ns1.firma.sk AAAA fd00::40 A 10.0.0.40 OPT
fd00::100  fd00::40      DNS         103 Standard query 0x9ef4 A ns1.firma.sk OPT
fd00::100  fd00::40      DNS         103 Standard query 0xaae3 AAAA ns1.firma.sk OPT
fd00::40   fd00::100     DNS         119 Standard query response 0x9ef4 A ns1.firma.sk A 10.0.0.40 OPT
fd00::40   fd00::100     DNS         131 Standard query response 0xaae3 AAAA ns1.firma.sk AAAA fd00::40 OPT
10.0.0.100 10.0.0.40     DNS          96 Standard query 0x57e9 PTR 1.18.200.199.in-addr.arpa OPT
10.0.0.40  10.0.0.100    DNS         126 Standard query response 0x57e9 PTR 1.18.200.199.in-addr.arpa PTR server1.firma.sk OPT
10.0.0.100 10.0.0.253    DNS         188 Standard query response 0xd662 PTR 1.18.200.199.in-addr.arpa PTR server1.firma.sk NS ns1.firma.sk
```

Obr. A.8: Zachytené správy vo Wiresharku

Otázka 6: Prečo sa v prípade použitia rekurzívneho resolvera zvýšil počet DNS správ? Aké správy boli navyše?
Otázka 7: Koľko správ sa prenesie, ak sa rekurzívneho resolvera spýtame na `test.lab.cz` znova? Aký je rozdiel v čase prekladu?
Otázka 8: Existuje pre doménu `firma.sk` MX záznam? Ak, áno, koľko ich je? Aké sú IP adresy mailových serverov? Majú prípadne tieto IP adresy PTR záznamy?
Otázka 9: Ako je nastavený flag „`aa`“ v odpovediach od rekurzívneho resolvera a prečo?

Internacionalizované doménové mená

V tejto časti si ukážeme, ako funguje preklad pomocou internacionalizovaných doménových mien (IDN - *Internationalized Domain Names*). Aby sa nemusel pre podporu IDN prekopávať celý systém DNS, tak sa vymyslelo, že Unicode znaky budeme ukladať v databázach vo formáte ACE (*ASCII Compatible Encoding*) pomocou tzv. Punycode. Príkladom môže byť napríklad reťazec „`háčkyčárky`“, ktorý by po prevedení vyzeral ako „`xn-hkyrky-ptac70bc`“, kde „`xn-`“ je prefix, podľa ktorého vieme, že prebehol takýto preklad. Podpora tohto systému ostáva na klientských staniciach a v dnešnej dobe predovšetkým na webových prehliadačoch. Program dig priamo IDN nepodporuje, môžeme si ale pomôcť programom idn2:

„`idn2 háčkyčárky`“.

Vidíme, že sme dostali vyššie spomenutý výsledok. Ďalej si skúsime preložiť doménové meno `háčkyčárky.lab.cz`. Keď to chceme zakomponovať do prekladu pomocou `dig`, použijeme príkaz:

„`dig @10.0.0.100 $(idn2 háčkyčárky.lab.cz)`“.

Tento príkaz ale hľadá iba A záznamy a také neboli nájdené, musíme ho upraviť tak, aby hľadal ľubovoľný záznam (viď príručku `dig -help`).

Otázka 10: Aký záznam sa skrýva pod doménovým menom `háčkyčárky.lab.cz`?

Zóna a zónové súbory

V tejto časti si ukážeme, ako vyzerá zónový súbor. Budeme sa venovať zóne `firma.sk`. Zónový súbor si môžeme prehliadnuť pomocou terminálu príkazom:

„`cat /dns-lab/server-firma_sk-a/bind/lib/db.firma.sk.zone`“,

alebo sa prihlásime do Webminu na `server-firma_sk-a` (odkazy na ploche). Klikneme na *Servers* a *BIND DNS Server*. Následne si otvoríme zónu `firma.sk`. Na tomto mieste je možné meniť rôzne parametre zóny a taktiež zobraziť zónový súbor kliknutím na *Edit Records File*. V oboch prípadoch by sme sa mali dostať k súboru, ako je znázornený vo výpise A.4.

Výpis A.4: Zónový súbor `db.firma.sk.zone`

```
$TTL 300 ; Default TTL
firma.sk.               IN      SOA     ns1.firma.sk. hostmaster.firma.sk. (
                                        100     ; Serial
                                        1800    ; Refresh
                                        900     ; Retry
                                        604800  ; Expire
```

```
                                        300 )   ; Negative Cache TTL
firma.sk.               IN      NS      ns1.firma.sk.
firma.sk.               IN      MX      5  mailin1.firma.sk.
firma.sk.               IN      MX      10 mailin2.firma.sk.
firma.sk.               IN      TXT     "Toto je firemna domena"
ns1.firma.sk.           IN      A       10.0.0.40
ns1.firma.sk.           IN      AAAA    fd00::40
server1.firma.sk.       IN      A       199.200.18.1
server1.firma.sk.       IN      AAAA    2001:1234:5678:90ab::1000
www.firma.sk.       600 IN      CNAME   server1.firma.sk.
webmail.firma.sk.   600 IN      CNAME   server1.firma.sk.
mailin1.firma.sk.       IN      A       199.200.18.2
mailin1.firma.sk.       IN      AAAA    2001:1234:5678:90ab::2000
mailin2.firma.sk.       IN      A       199.200.18.3
mailin2.firma.sk.       IN      AAAA    2001:1234:5678:90ab::3000
mail.firma.sk.          IN      A       199.200.18.4
mail.firma.sk.          IN      AAAA    2001:1234:5678:90ab::4000
mail.firma.sk.          IN      A       199.200.18.5
mail.firma.sk.          IN      AAAA    2001:1234:5678:90ab::5000
```

V tomto súbore je prednastavená hodnota TTL na 300 sekúnd, takže to nie je nutné definovať pre každý záznam. Môžeme to ale explicitne definovať, tak ako to je v tomto prípade pre CNAME záznamy. SOA záznam obsahuje niekoľko parametrov, ktoré sa týkajú zónových prenosov medzi primárnym a sekundárnym doménovým serverom. Je rozpísaný na niekoľko riadkov vďaka použitiu jednoduchých zátvoriek.

Rovnakým postupom si môžeme prezrieť aj iné zónové súbory. Napr. zónový súbor pre reverznú doménu `db.18.200.199.in-addr.arpa` vyzerá nasledovne:

Výpis A.5: Zónový súbor `db.18.200.199.in-addr.arpa.rev`

```
$TTL 300 ; Default TTL
18.200.199.in-addr.arpa.    IN  SOA ns1.firma.sk. hostmaster.firma.sk. (
                                    100           ; Serial
                                    1800          ; Refresh
                                    900           ; Retry
                                    604800        ; Expire
                                    300 )       ; Negative Cache TTL
18.200.199.in-addr.arpa.    IN  NS  ns1.firma.sk.
1.18.200.199.in-addr.arpa.  IN  PTR server1.firma.sk.
2.18.200.199.in-addr.arpa.  IN  PTR mailin1.firma.sk.
3.18.200.199.in-addr.arpa.  IN  PTR mailin2.firma.sk.
4.18.200.199.in-addr.arpa.  IN  PTR mail.firma.sk.
5.18.200.199.in-addr.arpa.  IN  PTR mail.firma.sk.
```

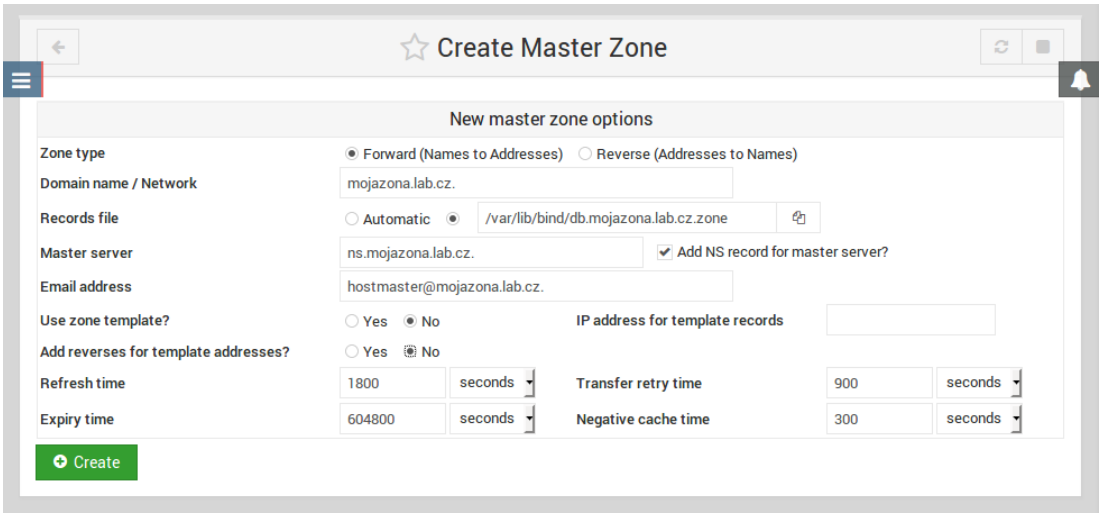
V tomto súbore vidíme (výpis A.5) iba SOA, NS a PTR záznamy.

Otázka 11: Aké údaje všeobecne obsahuje záznam SOA a načo slúžia? Vyhľadajte na internete.

Vytvorenie novej zóny

V tejto časti si vytvoríme novú zónu `mojazona.lab.cz` na serveri `server-lab_cz-a`. Môžeme to spraviť priamo vytvorením/upravením potrebných súborov alebo použi-

tím Webminu. Nasledujúci postup popisuje druhú možnosť. Prihlásime sa do Webminu daného servera a vojdeme do nastavení BIND-u. Ďalej klikneme na *Create master zone*. V zobrazenom formulári vyplníme údaje podľa obr. A.9.



Obr. A.9: Vytváranie novej zóny vo Webmine

Zónový súbor vytvorený pomocou Webminu nebude mať v Ubuntu nastavené práva na zápis používateľom `student`, ďalšie zmeny teda bude nutné vykonávať vo Webmine. Keď si zónový súbor otvoríme (`cat` v termináli, pomocou správcu súborov alebo vo Webmine), mal by vyzerať ako vo výpise A.6.

Výpis A.6: Zónový súbor `db.mojazona.lab.cz.zone`

```
$ttl 300
mojazona.lab.cz.   IN       SOA      ns.mojazona.lab.cz. hostmaster.mojazona.lab.cz. (
                                        1526808673
                                        1800
                                        900
                                        604800
                                        300 )
mojazona.lab.cz.   IN       NS       ns.mojazona.lab.cz.
```

Ak sa pozrieme do konfiguračného súboru `named.conf.local`, môžeme si všimnúť, že bola pridaná nová zóna (výpis A.7).

Výpis A.7: Konfiguračný súbor `named.conf.local` na `server-lab_cz`

```
zone "lab.cz" {
    type master;
    file "/var/lib/bind/db.lab.cz.zone";
};

zone "mojazona.lab.cz" {
    type master;
    file "/var/lib/bind/db.mojazona.lab.cz.zone";
};
```

V ďalšom kroku je potrebné oznámiť nadradenej zóne, že existuje nová práve pridaná zóna. Nadradená našej novej zóne je práve zóna `lab.cz`, do ktorej musíme vložiť NS záznam zhodný s tým, ktorý je v našom zónovom súbore, tzn:

„`mojazona.lab.cz. IN NS ns.mojazona.lab.cz.`“,

okrem toho musíme pridať aj tzv. glue záznam:

„`ns.mojazona.lab.cz. IN A 10.0.0.30`“,

a obdobne pre IPv6 adresu:

„`ns.mojazona.lab.cz. IN AAAA fd00::30`“.

Tieto úpravy môžeme postupne vykonať cez Webmin, alebo môžeme priamo upraviť zónový súbor `db.lab.cz.zone`. Vyššie spomenuté glue záznamy je tiež potrebné pridať do našej novo vytvorenej zóny, takže tieto tri záznamy budú zhodné pre obidve zóny. Okrem toho môžeme pridať do našej zóny ľubovoľný záznam, ktorý funkčnosť novej zóny overíme. Môže to byť napr. CNAME záznam odkazujúci na doménové meno mimo našej zóny:

„`mojcname.mojazona.lab.cz. IN CNAME test.lab.cz.`“.

Na záver je potrebné znova načítať konfiguráciu prípadne reštartovať BIND. Vo Webmine sa stačí vrátiť na základnú stránku konfigurácie BIND-u a v pravom hornom rohu kliknut na ikonu *Apply configuration*. V termináli stačí spustiť nasledujúci príkaz:

„`docker exec -it dns-lab.server-lab_cz rndc reload`“,

kde dostaneme odozvu `server reload successful`. Teraz môžeme našu novú zónu otestovať:

„`dig @10.0.0.100 mojcname.mojazona.lab.cz.`“,

kde by sme mali dostať IP adresu `1.2.3.4`. Overíme vo Wiresharku.

Otázka 12: Prečo sme museli pridať glue záznamy do nadradenej zóny?

Konfigurácia sekundárneho autoritatívneho servera

V poslednej časti tejto laboratórnej úlohy máme za cieľ nakonfigurovať sekundárny DNS server k už existujúcemu primárnemu serveru. Na toto zadanie využijeme servery `server-firma_sk-a` a `server-firma_sk-b`. Na našom sekundárnom serveri otvoríme súbor `named.conf.local`, ktorý by mal byť prázdny a pridáme doň nasledujúce riadky:

Výpis A.8: Konfiguračný súbor `named.conf.local` na `server-firma_sk-b`

```
zone "firma.sk" {
    type slave;
    masters { 10.0.0.40; };
    file "/var/cache/bind/db.firma.sk.zone";
};

zone "18.200.199.in-addr.arpa" {
    type slave;
    masters { 10.0.0.40; };
    file "/var/cache/bind/db.18.200.199.in-addr.arpa.rev";
};

zone "b.a.0.9.8.7.6.5.4.3.2.1.1.0.0.2.ip6.arpa" {
    type slave;
    masters { 10.0.0.40; };
    file "/var/cache/bind/db.b.a.0.9.8.7.6.5.4.3.2.1.1.0.0.2.ip6.arpa.rev";
};
```

Týmto sme zadefinovali zóny pre sekundárny server a nastavili adresu primárneho servera (master). Podobne upravíme súbor `named.conf.local` na `server-firma_sk-a`, kde doplníme ku každej zóne 2 riadky:

Výpis A.9: Konfiguračný súbor `named.conf.local` na `server-firma_sk-a`

```
zone "firma.sk" {
    type master;
    file "/var/lib/bind/db.firma.sk.zone";
    notify yes;
    allow-transfer { 10.0.0.41; };
};

zone "18.200.199.in-addr.arpa" {
    type master;
    file "/var/lib/bind/db.18.200.199.in-addr.arpa.rev";
    notify yes;
    allow-transfer { 10.0.0.41; };
};

zone "b.a.0.9.8.7.6.5.4.3.2.1.1.0.0.2.ip6.arpa" {
    type master;
    file "/var/lib/bind/db.b.a.0.9.8.7.6.5.4.3.2.1.1.0.0.2.ip6.arpa.rev";
    notify yes;
    allow-transfer { 10.0.0.41; };
};
```

Prvý server už obsahuje niekoľko zón (viď tab. A.1) a tak mu stačí zadefinovať aby notifikoval sekundárne doménové servery (podľa NS záznamov) a tiež aby mohol používať prenosy na dané servery. V našom prípade povoľujeme prenos len na IPv4 adrese. Následne je potrebné upraviť nadradené zóny (pridať záznamy), ktorých sa to týka, ako aj upravovanú zónu samotnú:

Výpis A.10: Úpravy zónových súborov

```
; db.firma.sk.zone
firma.sk.       IN      NS      ns2.firma.sk.
```

```
ns2.firma.sk.    IN       A      10.0.0.41
ns2.firma.sk.    IN       AAAA   fd00::41

; db.sk.zone
firma.sk.        IN       NS       ns2.firma.sk.
ns2.firma.sk.    IN       A      10.0.0.41
ns2.firma.sk.    IN       AAAA   fd00::41

; db.200.199.in-addr.arpa.zone.rev
18.200.199.in-addr.arpa.  IN  NS  ns2.firma.sk.

; db.18.200.199.in-addr.arpa.zone.rev
18.200.199.in-addr.arpa.  IN  NS  ns2.firma.sk.

; db.8.7.6.5.4.3.2.1.1.0.0.2.ip6.arpa
b.a.0.9.8.7.6.5.4.3.2.1.1.0.0.2.ip6.arpa.   IN  NS  ns2.firma.sk.

; db.b.a.0.9.8.7.6.5.4.3.2.1.1.0.0.2.ip6.arpa
b.a.0.9.8.7.6.5.4.3.2.1.1.0.0.2.ip6.arpa.   IN  NS  ns2.firma.sk.
```

Nakoniec je potrebné znovu načítať konfigurácie dotknutých serverov príkazom

„`docker exec -it dns-lab.<nazov_servera> rndc reload`“,

alebo môžeme využiť odkaz na ploche `Restart LAB1`, ktorý reštartuje všetky servery. Postupne by sme mali vo Wiresharku vidieť notifikácie o zmenách a množstvo AXFR prenosov.

A.2 Vzorové riešenie

Základný preklad DNS

Študent by mal postupne spustiť príkazy:

- „`dig @10.0.0.10 test.lab.cz.`“,
- „`dig @10.0.0.20 test.lab.cz. +norecurse`“,
- „`dig @10.0.0.30 test.lab.cz. +norecurse`“,
- „`dig @10.0.0.30 test.lab.cz. AAAA +norecurse`“,

prípadne verzie príkazov s IPv6 adresami (alebo ich kombináciou). Využiť mohol aj nástroj `nslookup`. Vo Wiresharku by mal mať zachytené DNS správy podobne ako na obr. A.10.

```
10.0.0.253 10.0.0.10    DNS          82 Standard query 0xffce A test.lab.cz OPT
10.0.0.10  10.0.0.253   DNS         149 Standard query response 0xffce A test.lab.cz NS a.ns.nic.cz A 10.0.0.20 AAAA fd00::20 OPT
10.0.0.253 10.0.0.20    DNS          82 Standard query 0xa87e A test.lab.cz OPT
10.0.0.20  10.0.0.253   DNS         144 Standard query response 0xa87e A test.lab.cz NS ns1.lab.cz A 10.0.0.30 AAAA fd00::30 OPT
10.0.0.253 10.0.0.30    DNS          82 Standard query 0x7c6a A test.lab.cz OPT
10.0.0.30  10.0.0.253   DNS         160 Standard query response 0x7c6a A test.lab.cz A 1.2.3.4 NS ns1.lab.cz A 10.0.0.30 AAAA fd00::30 OPT
10.0.0.253 10.0.0.30    DNS          82 Standard query 0xb26d AAAA test.lab.cz OPT
10.0.0.30  10.0.0.253   DNS         172 Standard query response 0xb26d AAAA test.lab.cz AAAA 2001:1ab::1234:5678 NS ns1.lab.cz A 10.0.0.30 AAAA fd00::30 OPT
```

Obr. A.10: Zachytené správy vo Wiresharku

Odpoveď 1: A záznam obsahuje IPv4 adresu `1.2.3.4`. Áno, obsahuje IPv6 adresu `2001:1ab::1234:5678`. Je potrebné pridať programu `dig` parameter `AAAA`.

Odpoveď 2: Koreňový server nie je nastavený ako rekurzívny DNS server resp. má vypnutú rekurziu.
Odpoveď 3: Bol použitý protokol UDP na porte 53.
Odpoveď 4: Veľkosti správ prvého dopytu boli 40 a 107 bajtov.

Reverzný preklad

Študent by mal postupne spustiť príkazy:

- „`dig @10.0.0.10 -x 199.200.18.1 +norecurse`“ alebo
- „`dig @10.0.0.10 1.18.200.199.in-addr.arpa. PTR +norecurse`“,
- „`dig @10.0.0.10 ns1.isp.sk. +norecurse`“,
- „`dig @10.0.0.25 ns1.isp.sk. +norecurse`“,
- „`dig @10.0.0.50 -x 199.200.18.1 +norecurse`“,
- „`dig @10.0.0.10 ns1.firma.sk. +norecurse`“,
- „`dig @10.0.0.25 ns1.firma.sk. +norecurse`“,
- „`dig @10.0.0.40 -x 199.200.18.1 +norecurse`“.

```
10.0.0.253 10.0.0.10    DNS          96 Standard query 0x2fdd PTR 1.18.200.199.in-addr.arpa OPT
10.0.0.10  10.0.0.253   DNS         120 Standard query response 0x2fdd PTR 1.18.200.199.in-addr.arpa NS ns1.isp.sk OPT
10.0.0.253 10.0.0.10    DNS          81 Standard query 0xbf89 A ns1.isp.sk OPT
10.0.0.10  10.0.0.253   DNS         145 Standard query response 0xbf89 A ns1.isp.sk NS a.tld.sk A 10.0.0.25 AAAA fd00::25 OPT
10.0.0.253 10.0.0.25    DNS          81 Standard query 0xf16e A ns1.isp.sk OPT
10.0.0.25  10.0.0.253   DNS         161 Standard query response 0xf16e A ns1.isp.sk A 10.0.0.50 NS a.tld.sk A 10.0.0.25 AAAA fd00::25 OPT
10.0.0.253 10.0.0.50    DNS          96 Standard query 0x0024 PTR 1.18.200.199.in-addr.arpa OPT
10.0.0.50  10.0.0.253   DNS         122 Standard query response 0x0024 PTR 1.18.200.199.in-addr.arpa NS ns1.firma.sk OPT
10.0.0.253 10.0.0.10    DNS          83 Standard query 0x7cca A ns1.firma.sk OPT
10.0.0.10  10.0.0.253   DNS         147 Standard query response 0x7cca A ns1.firma.sk NS a.tld.sk A 10.0.0.25 AAAA fd00::25 OPT
10.0.0.253 10.0.0.25    DNS          83 Standard query 0xd3d0 A ns1.firma.sk OPT
10.0.0.25  10.0.0.253   DNS         141 Standard query response 0xd3d0 A ns1.firma.sk NS ns1.firma.sk A 10.0.0.40 AAAA fd00::40 OPT
10.0.0.253 10.0.0.40    DNS          96 Standard query 0x983c PTR 1.18.200.199.in-addr.arpa OPT
10.0.0.40  10.0.0.253   DNS         188 Standard query response 0x983c PTR 1.18.200.199.in-addr.arpa PTR server1.firma.sk NS ns1.firma.sk
```

Obr. A.11: Zachytené správy vo Wiresharku

Odpoveď 5: Hľadané doménové meno je `server1.firma.sk`.

Rekurzívny resolver a využitie cache pamäte

Študent by mal postupne spustiť príkazy:

- „`dig @10.0.0.100 test.lab.cz.`“,
- „`dig @10.0.0.100 -x 199.200.18.1`“,
- „`dig @10.0.0.100 test.lab.cz.`“,
- „`dig @10.0.0.100 firma.sk MX`“,
- „`dig @10.0.0.100 -x 199.200.18.2`“,
- „`dig @10.0.0.100 -x 199.200.18.3`“,
- „`dig @10.0.0.100 -x 2001:1234:5678:90ab::2000`“,
- „`dig @10.0.0.100 -x 2001:1234:5678:90ab::3000`“.

Odpoveď 6: Navyše bola komunikácia resolvera s klientom (2 správy), zisťovanie informácií o koreňovej zóne (2 správy) a zisťovanie okrem A záznamov aj AAAA záznamov pre doménové servery (8 správ). Celkovo tak bolo prenesených 26 správ namiesto 14.
Odpoveď 7: Prenesú sa len dve správy (otázka a odpoveď) pretože rekurzívny resolver má túto informáciu uloženú v pamäti cache, kým neuplynie doba TTL. Čas prekladu rapídne klesne z niekoľkých desiatok milisekúnd na cca 1 ms.
Odpoveď 8: MX záznam pre doménu `firma.sk` existuje a je znázornený vo výpise A.11. PTR záznamy taktiež existujú a odkazujú na správne doménové názvy.
Odpoveď 9: Flag označujúcu autoritatívnu odpoveď nie je nastavený, pretože v našom prípade rekurzívny resolver nie je autoritatívny pre žiadnu doménu.

Výpis A.11: Výstup programu dig - dopyt na MX záznam

```
student@dns-lab:~$ dig @10.0.0.100 firma.sk MX
; <<>> DiG 9.10.3-P4-Ubuntu <<>> @10.0.0.100 firma.sk MX
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 39532
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 1, ADDITIONAL: 7

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;firma.sk.      IN  MX

;; ANSWER SECTION:
firma.sk.   300 IN  MX  5 mailin1.firma.sk.
firma.sk.   300 IN  MX  10 mailin2.firma.sk.

;; AUTHORITY SECTION:
firma.sk.   300 IN  NS  ns1.firma.sk.

;; ADDITIONAL SECTION:
mailin1.firma.sk. 300 IN  A 199.200.18.2
mailin1.firma.sk. 300 IN  AAAA  2001:1234:5678:90ab::2000
mailin2.firma.sk. 300 IN  A 199.200.18.3
mailin2.firma.sk. 300 IN  AAAA  2001:1234:5678:90ab::3000
ns1.firma.sk.   300 IN  A 10.0.0.40
ns1.firma.sk.   300 IN  AAAA  fd00::40

;; Query time: 33 msec
;; SERVER: 10.0.0.100#53(10.0.0.100)
;; MSG SIZE  rcvd: 235
```

Internacionalizované doménové mená

Študent by mal postupne spustiť príkazy:

- „`idn2 háčkyčárky`“,
- „`dig @10.0.0.100 $(idn2 háčkyčárky.lab.cz)`“,

- „`dig @10.0.0.100 $(idn2 háčkyčárky.lab.cz) ANY)`“.

Odpoveď 10: Po použití parametra `ANY` by sa mal študent dostať k TXT záznamu s obsahom `"Toto je ten spravny zaznam"`

Zóna a zónové súbory

Odpoveď 11: Záznam SOA začína názvom primárneho menného servera, pokračuje emailovou adresou správcu zóny, kde sa @ nahradzuje bodkou. Za týmito údajmi sa tam nachádza 5 číselných údajov. Serial je verzia zónového súboru. Refresh udáva po akom čase sa bude kontrolovať verzia zónového súboru na primárnom serveri po predchádzajúcej kontrole alebo načítaní zóny. Retry udáva časový úsek po ktorom v prípade neúspešného zisťovania verzie zónového súboru to skúsi znova. Expire je maximálny čas platnosti zóny v prípade neúspechu zistiť sériové číslo. Posledný parameter viac krát v minulosti menil svoj význam. V súčasnosti znamená Negative Cache TTL a určuje dobu uchovania informácií o neexistujúcom zázname v pamati cache.

Vytvorenie novej zóny

Študent by mal úspešne vytvoriť novú zónu `mojazona.lab.cz` podľa podrobného postupu. Taktiež by mal vedieť overiť funkčnosť novej zóny.

Odpoveď 12: Glue záznamy slúžia nadradenej zóne na to, aby vedela, aké autoritatívne servery obsluhujú danú zónu.

Konfigurácia sekundárneho autoritatívneho servera

```
10.0.0.40  10.0.0.41   DNS      142 Zone change notification 0xbcf8 SOA 18.200.199.in-addr.arpa SOA ns1.firma.sk
10.0.0.41  10.0.0.40   ICMP     170 Destination unreachable (Port unreachable)
fd00::40   fd00::41    DNS      162 Zone change notification 0xf5c8 SOA 18.200.199.in-addr.arpa SOA ns1.firma.sk
fd00::41   fd00::40    ICMPv6   210 Destination Unreachable (Port unreachable)
10.0.0.40  10.0.0.41   DNS      159 Zone change notification 0x671d SOA b.a.0.9.8.7.6.5.4.3.2.1.1.0.0.2.ip6.arpa SOA ns1.firma.sk
10.0.0.41  10.0.0.40   ICMP     187 Destination unreachable (Port unreachable)
10.0.0.40  10.0.0.41   DNS      119 Zone change notification 0x307a SOA firma.sk SOA ns1.firma.sk
10.0.0.41  10.0.0.40   ICMP     147 Destination unreachable (Port unreachable)
fd00::40   fd00::41    DNS      139 Zone change notification 0xa30c SOA firma.sk SOA ns1.firma.sk
fd00::41   fd00::40    ICMPv6   187 Destination Unreachable (Port unreachable)
fd00::40   fd00::41    DNS      179 Zone change notification 0x9422 SOA b.a.0.9.8.7.6.5.4.3.2.1.1.0.0.2.ip6.arpa SOA ns1.firma.sk
fd00::41   fd00::40    ICMPv6   227 Destination Unreachable (Port unreachable)
10.0.0.41  10.0.0.40   DNS       94 Standard query 0x2c31 SOA 18.200.199.in-addr.arpa OPT
10.0.0.40  10.0.0.41   DNS      273 Standard query response 0x2c31 SOA 18.200.199.in-addr.arpa SOA ns1.firma.sk NS ns2.firma.sk NS
10.0.0.41  10.0.0.40   DNS      109 Standard query 0xca67 AXFR 18.200.199.in-addr.arpa
10.0.0.40  10.0.0.41   DNS      345 Standard query response 0xca67 AXFR 18.200.199.in-addr.arpa SOA ns1.firma.sk NS ns1.firma.sk N
10.0.0.41  10.0.0.40   DNS       79 Standard query 0x5aaf SOA firma.sk OPT
10.0.0.41  10.0.0.40   DNS      111 Standard query 0xa0a9 SOA b.a.0.9.8.7.6.5.4.3.2.1.1.0.0.2.ip6.arpa OPT
10.0.0.40  10.0.0.41   DNS      250 Standard query response 0x5aaf SOA firma.sk SOA ns1.firma.sk NS ns1.firma.sk NS ns2.firma.sk A
10.0.0.40  10.0.0.41   DNS      290 Standard query response 0xa0a9 SOA b.a.0.9.8.7.6.5.4.3.2.1.1.0.0.2.ip6.arpa SOA ns1.firma.sk N
10.0.0.41  10.0.0.40   DNS       94 Standard query 0x92fe AXFR firma.sk
10.0.0.41  10.0.0.40   DNS      126 Standard query 0x9454 AXFR b.a.0.9.8.7.6.5.4.3.2.1.1.0.0.2.ip6.arpa
10.0.0.40  10.0.0.41   DNS      657 Standard query response 0x92fe AXFR firma.sk SOA ns1.firma.sk NS ns1.firma.sk NS ns2.firma.sk
10.0.0.40  10.0.0.41   DNS      416 Standard query response 0x9454 AXFR b.a.0.9.8.7.6.5.4.3.2.1.1.0.0.2.ip6.arpa SOA ns1.firma.sk
```

Obr. A.12: Zachytené správy vo Wiresharku

Študent by mal úspešne nakonfigurovať sekundárny doménový server k primárnemu `server-firma_sk-a`. Notifikovanie o zmenách a AXFR prenosy by mal študent zachytiť vo Wiresharku, viď obr. A.12.

B LABORATÓRNA ÚLOHA 2

B.1 Návod

B.1.1 Ciele laboratórnej úlohy

- zoznámenie sa s DNSSEC,
- nastavenie rekurzívneho resolvera,
- overenie DNSSEC validácie,
- reťazec dôvery v DNSSEC,
- sledovanie správania transportnej vrstvy,
- porovnanie výkonu verejných DNS serverov.

B.1.2 Teoretický úvod

DNSSEC

DNSSEC (*Domain Name System Security Extensions*) je rozšírenie systému DNS, pomocou ktorého vieme zabezpečiť pravosť získaných záznamov.[8] Funguje na princípe podpisovania záznamov. Nejde teda o šifrovanie samotných záznamov. Rekurzívny resolver v prípade zabezpečenej domény má možnosť overiť autenticitu získaných informácií. Toto zabezpečenie teda predpokladá bezpečný kanál medzi rekurzívnym resolverom a klientom. Je tak vhodné používať resolver, ktorý je umiestnený v lokálnej sieti, alebo ešte lepšie priamo v klientovi. DNSSEC pridáva nové typy zdrojových viet (RR):[2, s. 68]

- **DNSKEY** (*DNS Key record*) - obsahuje verejný kľúč, ktorý slúži na overovanie podpisov.
- **RRSIG** (*Resource Record Signature*) - obsahuje podpis pre konkrétny záznam. Tento podpis je overovaný resolvermi pomocou verejného kľúča v zázname DNSKEY.
- **DS** (*Delegation Signer*) - odtlačok verejného kľúča, ktorý sa vkladá do nadradenej domény, kde je podpísaný kľúčom nadradenej domény.
- **NSEC** (*Next Secure record*) - obsahuje odkaz na najbližší záznam (z pohľadu abecedného poradia), ktorý sa využíva v prípade odpovede na neexistujúci záznam (NXDOMAIN).
- **NSEC3** (*Next Secure record version 3*) - vylepšuje NSEC, keď namiesto reálneho mena obsahuje iba jeho hash. To zabraňuje prehľadávaniu celej zóny, čo umožňovali práve NSEC záznamy.

DNSSEC využíva pri podpisovaní asymetrickú kryptografiu, tzn. že využíva privátny kľúč na podpisovanie záznamov a verejný kľúč na validovanie podpisu. V

súčasnosti sa využívajú dva typy kľúčov, a to KSK (*key signing keys*) a ZSK (*zone signing keys*). KSK bývajú väčšie a slúžia na podpis ostatných kľúčov a teda aj ZSK, ktoré sú spravidla menšie, a slúžia na podpis ostatných záznamov. Týmto rozdelením sa zjednodušuje výmena kľúčov (KSK nie je potrebné meniť tak často), a tiež týmto redukujeme veľkosť RRSIG/DNSKEY záznamov.[10]

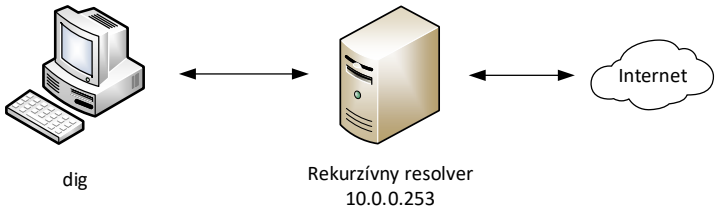
Základom overenia je reťazec dôvery, ktorý sa skladá z prepojených DS a DNSKEY záznamov, začínajúci vždy v koreňovej doméne, ktorej bod dôvery (trust anchor) je obsiahnutý v DNS serveroch ako súčasť programu, prípadne sa dá stiahnuť zo stránok organizácie IANA[9]. DNSSEC validácia je možná len vtedy, keď je tento reťazec kompletný až po dopytovanú doménu.

B.1.3 Laboratórne prostredie

Laboratórna úloha je pripravená v prostredí operačného systému Ubuntu Desktop 16.04 virtualizovaného na platforme Oracle VM VirtualBox. Prihlasovacie údaje do systému sú `student:student`. Zmeny vykonané vo virtualizovanom OS sú uchovávané len do reštartu virtuálneho stroja. Tento systém obsahuje program `dig` (terminál), Wireshark (GUI) a niekoľko predkonfigurovaných inštancií DNS serverov (BIND). Z hostiteľského OS (Windows) je možné využiť nástroj príkazového riadku `nslookup`. Na pracovnej ploche Ubuntu nájdete odkazy na tri skripty. `Start LAB1`, `Start LAB2` na spustenie jednotlivých laboratórnych úloh a `Restart LAB1` na reštartovanie DNS serverov 1. laboratórnej úlohy so zachovaním konfigurácie. Skript spustíte dvojklikom a následne vyberiete možnosť *Run in Terminal*. Prístup na webové rozhranie Webmin je možný pomocou adresy v tvare `https://<ip_adresa>:10000` alebo pomocou odkazov na ploche. Hlásenie o nedôveryhodnom (self signed) certifikáte ignorujeme. Prihlasovacie údaje do Webminu sú `root:password`. Lokálne IP adresy pre virtualizované Ubuntu sú `10.0.0.253, fd00::253`, pre hostiteľský OS Windows to sú `10.0.0.254, fd00::254`. Okrem webového rozhrania je možné upravovať konfiguračné súbory jednotlivých inštancií BIND. Tie sa nachádzajú v umiestnení `/dns-lab/<nazov_servera>/bind` (odkaz na ploche). V tomto priečinku sú podpriečinky `etc` pre `/etc/bind`, `lib` pre `/var/lib/bind` a `cache` pre `/var/cache/bind`.

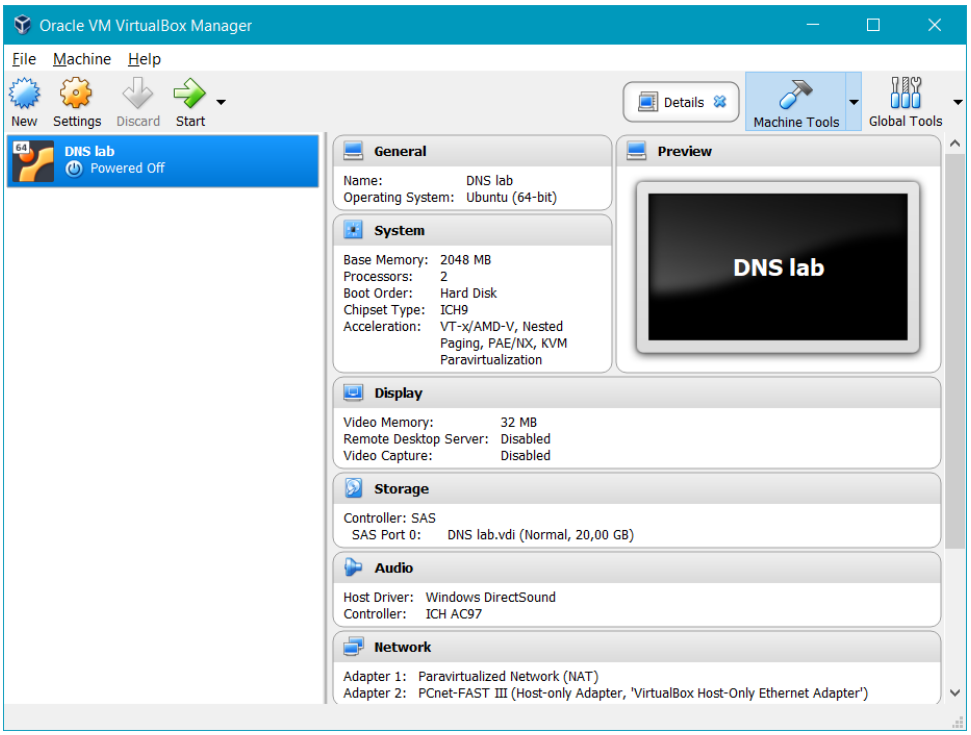
B.1.4 Počiatočná konfigurácia

V tejto úlohe je k dispozícii jeden server BIND v predvolenej konfigurácii a s prístupom na internet. Samotný server je pripojený do virtuálnej siete `docker0` (`172.17.0.0/16`), má ale publikované porty 53 (TCP a UDP) a 10000 (TCP) na IP adrese 10.0.0.253.



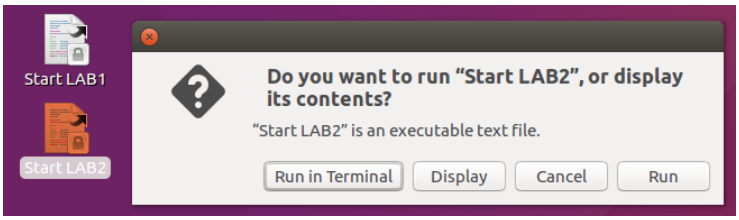
Obr. B.1: Logická topológia laboratórnej siete

B.1.5 Postup

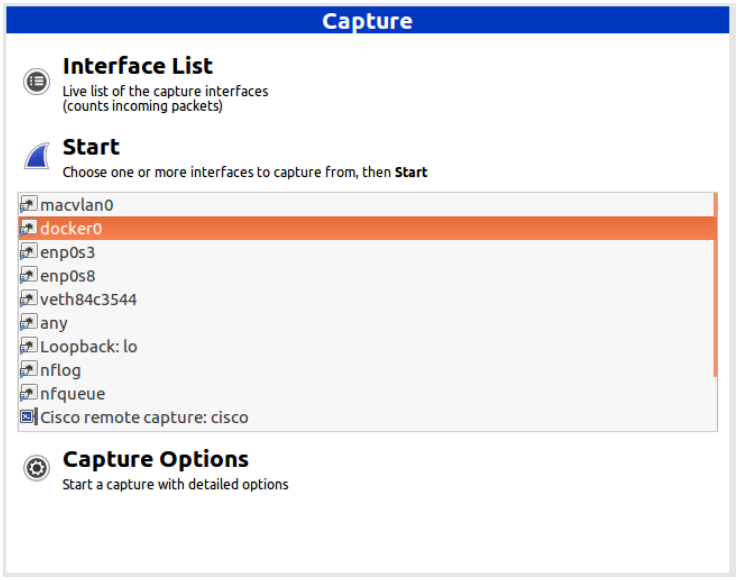


Obr. B.2: Základné okno programu Oracle VM VirtualBox

Spustíme program Oracle VM VirtualBox a v hlavnom okne vyberieme a spustíme virtuálny stroj (VM) s názvom `DNS lab` (obr. B.2). Po prihlásení pod účtom `student` (heslo `student`) spustíme skript `Start LAB2` nachádzajúci sa na pracovnej ploche (obr. B.3) a počkáme, kým nezmizne terminálové okno. Pre analyzovanie DNS spustíme vo virtuálnom systéme Wireshark a aktivujeme odchytávanie paketov na rozhraní `docker0` (obr. B.4). Potom aktivujeme filter „`dns`“ a potvrdíme tlačidlom *Apply* (obr. B.5).



Obr. B.3: Spustenie skriptu `Start Lab2`



Obr. B.4: Výber rozhrania na zachytávanie paketov vo Wiresharku

Nastavenie rekurzívneho resolvera

V tejto časti úlohy nastavíme BIND ako rekurzívny resolver s ukladaním do cache pamäte. Otvoríme si konfiguračný súbor `named.conf.options` (vo Webmine alebo priamo), ktorý vyzerá nasledovne:

Výpis B.1: Konfiguračný súbor `named.conf.options`

```
options {
  directory "/var/cache/bind";
  dnssec-validation auto;
  auth-nxdomain no;    # conform to RFC1035
  listen-on-v6 { any; };
};
```

Najprv na serveri vypneme validáciu DNSSEC, tzn. `auto` zmeníme na `no`. Potom povolíme rekurziu nastavením:

Výpis B.2: Zapnutie rekurzie v `named.conf.options`

```
allow-recursion { any; };
```

Obr. B.5: Nastavenie filtra vo Wiresharku pre zobrazenie DNS správ

```
allow-query { any; };
allow-query-cache { any; };
```

Nasledujúcim príkazom načítame novú konfiguráciu:

„`docker exec -it dns-lab.bind rndc reload`“.

Funkčnosť nastavenia si môžeme overiť prekladom:

„`dig @10.0.0.253 vutbr.cz`“.

Takisto skúsme preložiť doménové meno `www.rhybar.cz`:

„`dig @10.0.0.253 www.rhybar.cz`“.

Mali by sme sa dostať bez problémov k A záznamu daného doménového mena:

Výpis B.3: Výstup programu dig - dopyt na A záznam

```
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 39559
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 3, ADDITIONAL: 7

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.rhybar.cz.      IN   A

;; ANSWER SECTION:
www.rhybar.cz.     465 IN   A 217.31.205.55

;; AUTHORITY SECTION:
rhybar.cz.     465 IN   NS  b.ns.nic.cz.
rhybar.cz.     465 IN   NS  a.ns.nic.cz.
rhybar.cz.     465 IN   NS  d.ns.nic.cz.

;; ADDITIONAL SECTION:
a.ns.nic.cz.     1665   IN   A 194.0.12.1
a.ns.nic.cz.     1665   IN   AAAA  2001:678:f::1
b.ns.nic.cz.     1665   IN   A 194.0.13.1
b.ns.nic.cz.     1665   IN   AAAA  2001:678:10::1
d.ns.nic.cz.     1665   IN   A 193.29.206.1
d.ns.nic.cz.     1665   IN   AAAA  2001:678:1::1

;; Query time: 0 msec
;; SERVER: 10.0.0.253#53(10.0.0.253)

;; MSG SIZE  rcvd: 245
```

Otázka 1: Kam smeruje náš resolver dopyty na doménové záznamy, keď nemá nastaveného žiadneho forwardera?

Aktivácia DNSSEC validácie

V tejto časti zapneme validáciu DNSSEC. Stačí, ak nastavíme `dnssec-validation` naspať na `auto`. V tomto prípade bude náš súbor vyzerať nasledovne:

Výpis B.4: Konfiguračný súbor `named.conf.options`

```
options {
    directory "/var/cache/bind";
    dnssec-validation auto;
    allow-recursion { any; };
    allow-query { any; };
    allow-query-cache { any; };
    auth-nxdomain no;    # conform to RFC1035
    listen-on-v6 { any; };
};
```

Možnosť `auto` je v podstate rovnaká ako `yes`, s tým rozdielom, že v prvom prípade si bude BIND automaticky aktualizovať bod dôvery (trust anchor). V druhom sa o to musíme starať manuálne. Aktuálnu verziu si môžeme pozrieť v súbore `bind.keys`:

„`cat /dns-lab/bind/bind/etc/bind.keys`“.

Teraz môžeme načítať novú konfiguráciu a otestovať správanie resolvera. Skúsme znova preložiť doménové mená ako v predošlej časti. Ak sme postupovali správne, dopyt na druhé zadané meno by nám malo ostať nezodpovedané so statusom `SERVFAIL`. Táto doména je totiž zle podpísaná čo náš validujúci resolver vyhodnotil správne a vrátil nám chybu. Vo Wiresharku si môžeme pozrieť, že resolver A záznam dostal, ale neposlal ho ďalej. Bohužiaľ z takého všeobecného popisu chyby nemôžeme presne určiť, či je chyba v DNSSEC. Môžeme ale spustiť program dig s parametrom `+cdflag`. Príznak `CD` znamená Checking Disabled, tzn. že nevyužívame validáciu DNSSEC. Po takomto overení už vieme, že je problém v DNSSEC.

Výpis B.5: Výstup programu dig - dopyt na A záznam

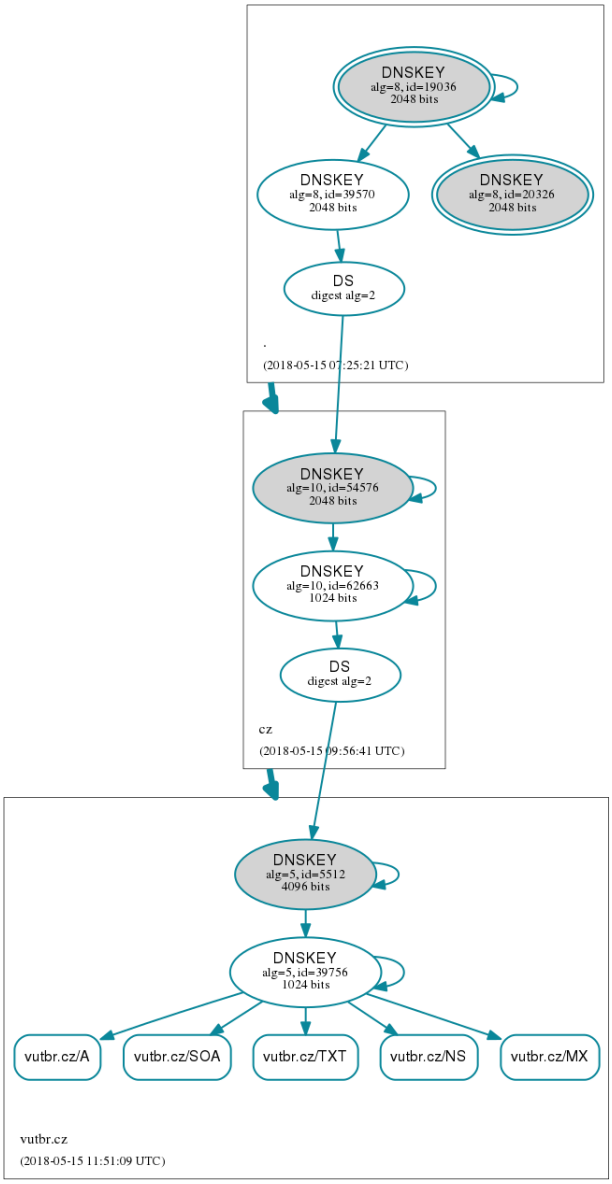
```
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 17196
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.rhybar.cz.     IN  A
```

```
;; Query time: 195 msec
;; SERVER: 10.0.0.253#53(10.0.0.253)
;; MSG SIZE  rcvd: 42
```

Otázka 2: Kde na internete môžeme nájsť aktuálne body dôvery pre koreňovú doménu?

Reťazec dôvery v DNSSEC

V tejto úlohe si pomocou webovej aplikácie DNSViz (`http://dnsviz.net/`) zobrazte reťazec dôvery pre doménové mená `vutbr.cz` a `www.rhybar.cz`.



Obr. B.6: DNSViz - reťazec dôvery doménového mena `vutbr.cz`

Otázka 3: Aký problém s reťazcom dôvery je v doméne `rhybar.cz`?

Transportná vrstva a DNSSEC

V tejto úlohe vyskúšame aký protokol transportnej vrstvy sa používa a kedy. Otvoríme program Wireshark a pozrieme sa, či bol použitý protokol UDP alebo TCP a skúsime zistiť, od čoho to záviselo. Ak sme mali spustený Wireshark od začiatku, máme zachytených dostatok paketov. V opačnom prípade môžeme vymazať cache nášho resolvera:

„`docker exec -it dns-lab.bind rndc flush`“,

a následne spustíme preklad rôznych domén, či už s DNSSEC-om alebo bez.

Otázka 4: Aký typ záznamu je najčastejšie prenášaný pomocou TCP? Aká je hraničná veľkosť kedy sa prepne na TCP?

Porovnanie výkonu verejných DNS serverov

Pre túto časť úlohy si vo Windowse stiahnite program Domain Name Server Benchmark zo stránky `https://www.grc.com/dns/benchmark.htm`. Po stiahnutí môžeme program spustiť a v záložke *Nameservers* necháme len servery, ktoré sú nastavené v systéme, niektoré známe verejné resolvery napr. `1.1.1.1`, `8.8.8.8`, `208.67.222.222` a taktiež náš virtuálny server, čiže `10.0.0.253`.

Otázka 5: Prečo majú niektoré servery dlhší priemerný čas získania nacachovaných záznamov?

B.2 Vzorové riešenie

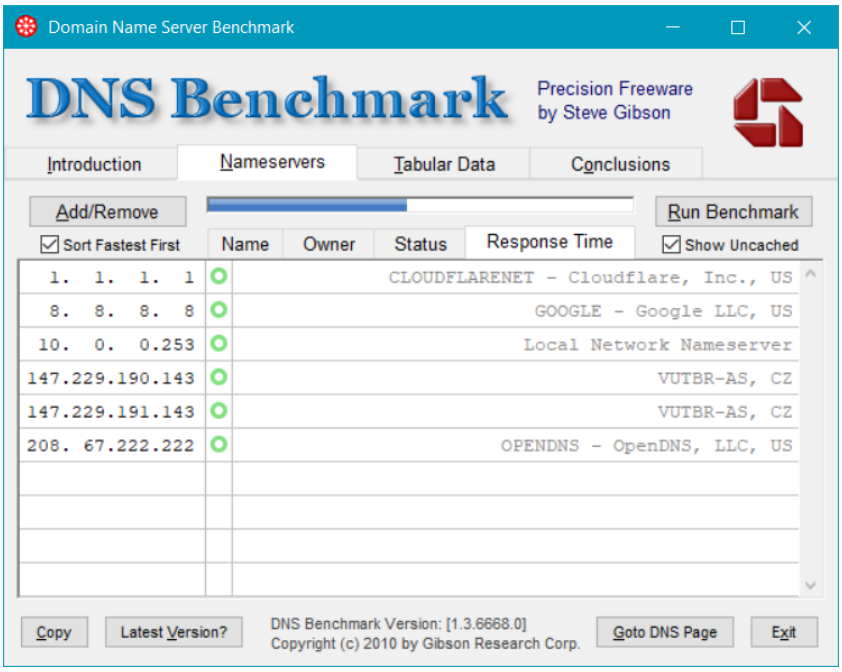
Nastavenie rekurzívneho resolvera

Odpoveď 1: Používa tzv. root hints, čo je malý súbor s IP adresami všetkých trinástich (nie fyzicky) koreňových serverov. A pomocou týchto sa rekurzívne pýta ďalej.

Aktivácia DNSSEC validácie

Výpis B.6: Výstup programu dig - dopyt na A záznam s príznakom `CD`

```
; (1 server found)
;; global options: +cmd
;; Got answer:
```



Obr. B.7: DNS Benchmark - nastavenie serverov

```
10.0.0.253 172.17.0.2   DNS          84 Standard query 0x50cc A www.rhybar.cz OPT
172.17.0.2 193.29.206.1 DNS          84 Standard query 0x0770 A www.rhybar.cz OPT
193.29.206.172.17.0.2   DNS         481 Standard query response 0x0770 A www.rhybar.cz NS a.ns.nic.cz NS b.ns.nic.cz NS d.ns
172.17.0.2 194.0.13.1   DNS          84 Standard query 0xd8ac A www.rhybar.cz OPT
194.0.13.1 172.17.0.2   DNS         493 Standard query response 0xd8ac A www.rhybar.cz A 217.31.205.55 RRSIG NS b.ns.nic.cz
172.17.0.2 194.0.12.1   DNS          80 Standard query 0xa7e5 DNSKEY rhybar.cz OPT
194.0.12.1 172.17.0.2   DNS         477 Standard query response 0xa7e5 DNSKEY rhybar.cz NS a.ns.nic.cz NS b.ns.nic.cz NS d.ns
172.17.0.2 193.29.206.1 DNS          80 Standard query 0xaf8c DNSKEY rhybar.cz OPT
193.29.206.172.17.0.2   DNS         477 Standard query response 0xaf8c DNSKEY rhybar.cz NS a.ns.nic.cz NS b.ns.nic.cz NS d.ns
172.17.0.2 194.0.13.1   DNS          80 Standard query 0x2300 DNSKEY rhybar.cz OPT
194.0.13.1 172.17.0.2   DNS        1118 Standard query response 0x2300 DNSKEY rhybar.cz DNSKEY DNSKEY DNSKEY RRSIG RRSIG OPT
172.17.0.2 10.0.0.253   DNS          84 Standard query response 0x50cc Server failure A www.rhybar.cz OPT
```

Obr. B.8: Zachytené správy vo Wiresharku

```
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 23490
;; flags: qr rd ra cd; QUERY: 1, ANSWER: 1, AUTHORITY: 3, ADDITIONAL: 7

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.rhybar.cz.      IN   A

;; ANSWER SECTION:
www.rhybar.cz.     147 IN   A 217.31.205.55

;; AUTHORITY SECTION:
rhybar.cz.     3147   IN   NS   d.ns.nic.cz.
rhybar.cz.     3147   IN   NS   a.ns.nic.cz.
rhybar.cz.     3147   IN   NS   b.ns.nic.cz.

;; ADDITIONAL SECTION:
a.ns.nic.cz.     3147   IN   A 194.0.12.1
a.ns.nic.cz.     3147   IN   AAAA   2001:678:f::1
b.ns.nic.cz.     3147   IN   A 194.0.13.1
b.ns.nic.cz.     3147   IN   AAAA   2001:678:10::1
```

```
d.ns.nic.cz.     3147   IN   A 193.29.206.1
d.ns.nic.cz.     3147   IN   AAAA   2001:678:1::1

;; Query time: 0 msec
;; SERVER: 10.0.0.253#53(10.0.0.253)
;; MSG SIZE  rcvd: 245
```

Odpoveď 2: Aktuálne body dôvery (trust anchors) nájdeme na `https://data.iana.org/root-anchors/root-anchors.xml`.

Reťazec dôvery v DNSSEC

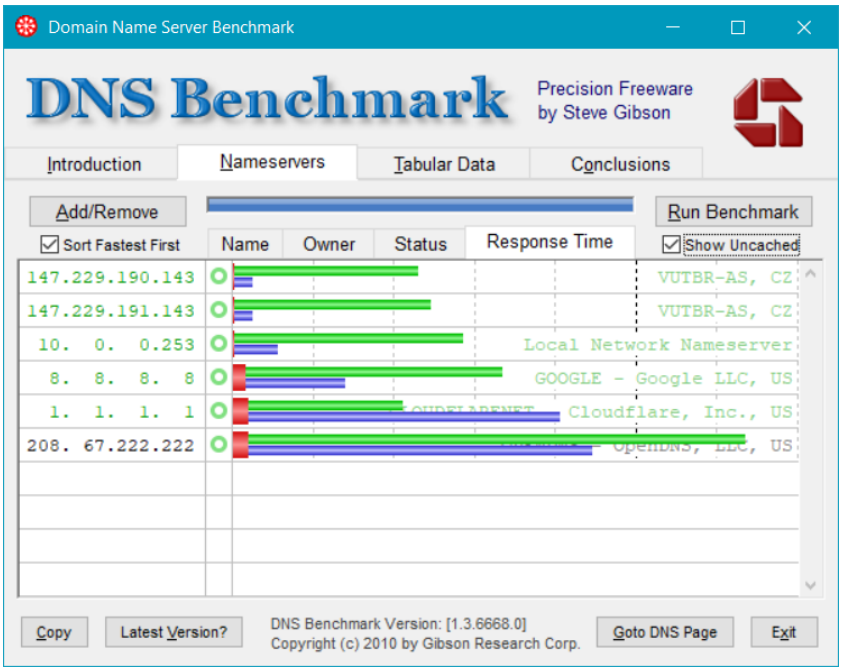
Odpoveď 3: Problém je v tom, že v nadradenej doméne `cz` odkazuje DS záznam na iný DNSKEY záznam, ako ten, ktorým sa reálne podpisujú záznamy.

Transportná vrstva a DNSSEC

Odpoveď 4: Najčastejšie to je DNSKEY. Hraničná veľkosť by mala byť 4096 bajtov, predtým to bolo 512 bajtov.

Porovnanie výkonu verejných DNS serverov

V nasledujúcom obrázku vidno výsledok porovnania rýchlosti odozvy zadaných DNS serverov.



Obr. B.9: DNS Benchmark - výsledky

Odpoveď 5: Pretože servery s krátkym časom odozvy sú veľmi blízko resp. v našej lokálnej sieti.

C SKRIPTY A KONFIGURAČNÉ SÚBORY

Výpis C.1: Skript na nastavenie IP adries (`dns_lab-set_network.sh`)

```
#!/bin/bash
sudo ip link add macvlan0 link enp0s8 type macvlan mode bridge
sudo ip address add 10.0.0.253/24 dev macvlan0
sudo ip address add fd00:0:0:0::253/64 dev macvlan0
sudo ifconfig macvlan0 up
```

Výpis C.2: Skript pre vytváranie kontajnerov (`dns_lab1-docker-run.sh`)

```
#!/bin/bash
echo "Creating dns-lab containers"
docker run --name dns-lab.server-root-a -h server-root-a \
    -itd --network macvlan-net --ip 10.0.0.10 --ip6 fd00::10 \
    --volume /dns-lab/server-root-a:/data --cap-add=NET_ADMIN dns-lab/bind:1.0
docker run --name dns-lab.server-root-b -h server-root-b \
    -itd --network macvlan-net --ip 10.0.0.11 --ip6 fd00::11 \
    --volume /dns-lab/server-root-b:/data --cap-add=NET_ADMIN dns-lab/bind:1.0
docker run --name dns-lab.server-cz -h server-cz \
    -itd --network macvlan-net --ip 10.0.0.20 --ip6 fd00::20 \
    --volume /dns-lab/server-cz:/data --cap-add=NET_ADMIN dns-lab/bind:1.0
docker run --name dns-lab.server-sk -h server-sk \
    -itd --network macvlan-net --ip 10.0.0.25 --ip6 fd00::25 \
    --volume /dns-lab/server-sk:/data --cap-add=NET_ADMIN dns-lab/bind:1.0
docker run --name dns-lab.server-lab_cz -h server-lab_cz \
    -itd --network macvlan-net --ip 10.0.0.30 --ip6 fd00::30 \
    --volume /dns-lab/server-lab_cz:/data --cap-add=NET_ADMIN dns-lab/bind:1.0
docker run --name dns-lab.server-firma_sk-a -h server-firma_sk-a \
    -itd --network macvlan-net --ip 10.0.0.40 --ip6 fd00::40 \
    --volume /dns-lab/server-firma_sk-a:/data --cap-add=NET_ADMIN dns-lab/bind:1.0
docker run --name dns-lab.server-firma_sk-b -h server-firma_sk-b \
    -itd --network macvlan-net --ip 10.0.0.41 --ip6 fd00::41 \
    --volume /dns-lab/server-firma_sk-b:/data --cap-add=NET_ADMIN dns-lab/bind:1.0
docker run --name dns-lab.server-isp -h server-isp \
    -itd --network macvlan-net --ip 10.0.0.50 --ip6 fd00::50 \
    --volume /dns-lab/server-isp:/data --cap-add=NET_ADMIN dns-lab/bind:1.0
docker run --name dns-lab.recursive-resolver -h recursive-resolver \
    -itd --network macvlan-net --ip 10.0.0.100 --ip6 fd00::100 \
    --volume /dns-lab/recursive-resolver:/data --cap-add=NET_ADMIN dns-lab/bind:1.0

docker exec -it dns-lab.server-root-a tc qdisc add dev eth0 root netem delay 10ms
docker exec -it dns-lab.server-root-b tc qdisc add dev eth0 root netem delay 10ms
docker exec -it dns-lab.server-cz tc qdisc add dev eth0 root netem delay 10ms
docker exec -it dns-lab.server-sk tc qdisc add dev eth0 root netem delay 10ms
docker exec -it dns-lab.server-lab_cz tc qdisc add dev eth0 root netem delay 10ms
docker exec -it dns-lab.server-firma_sk-a \
    tc qdisc add dev eth0 root netem delay 10ms
docker exec -it dns-lab.server-firma_sk-b \
    tc qdisc add dev eth0 root netem delay 10ms
docker exec -it dns-lab.server-isp tc qdisc add dev eth0 root netem delay 10ms
```

Výpis C.3: Skript pre vytváranie kontajnerov (`dns_lab2-docker-run.sh`)

```
#!/bin/bash
echo "Creating dns-lab containers"
docker run --name dns-lab.bind -h bind -itd -p 10.0.0.253:53:53/tcp \
-p 10.0.0.253:53:53/udp -p 10.0.0.253:10000:10000/tcp \
-v /dns-lab/bind:/data --cap-add=NET_ADMIN dns-lab/bind:1.0
```

Výpis C.4: Skript na načítanie LAB1 (`dns_lab1-start.sh`)

```
#!/bin/bash
echo "stopping dns-lab containers"
docker stop $(docker ps -q)
echo "Removing dns-lab containers"
docker rm $(docker ps -a -q)
rm -rfv /dns-lab/*
cp -R /dns-lab-defaults/dns-lab-1/. /dns-lab/
/scripts/dns_lab1-docker-run.sh
echo "Done"
sleep 1s
```

Výpis C.5: Skript na načítanie LAB2 (`dns_lab2-start.sh`)

```
#!/bin/bash
echo "stopping dns-lab containers"
docker stop $(docker ps -q)
echo "Removing dns-lab containers"
docker rm $(docker ps -a -q)
rm -rfv /dns-lab/*
cp -R /dns-lab-defaults/dns-lab-2/. /dns-lab/
/scripts/dns_lab2-docker-run.sh
echo "Done"
sleep 1s
```

Výpis C.6: Skript na vyčistenie kontajnerov (`dns_lab-clean.sh`)

```
#!/bin/bash
docker stop $(docker ps -q)
docker rm $(docker ps -a -q)
rm -rfv /dns-lab/*
echo "Done"
sleep 1s
```

Výpis C.7: Súbor Root Hints (`root.hints`)

```
.                    3600000     NS      A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET.  3600000     A       10.0.0.10
A.ROOT-SERVERS.NET.  3600000     AAAA    fd00::10
.                    3600000     NS      B.ROOT-SERVERS.NET.
B.ROOT-SERVERS.NET.  3600000     A       10.0.0.11
B.ROOT-SERVERS.NET.  3600000     AAAA    fd00::11
```

Výpis C.8: Zónový súbor koreňovej zóny (`db.root.zone`)

```
$TTL 300 ; Default TTL
.                    IN  SOA     a.root-servers.net. nstld.verisign-grs.com. (
                                 100      ; Serial
                                 1800     ; Refresh
                                 900      ; Retry
                                 604800   ; Expire
                                 300 )    ; Negative Cache TTL
.                    IN  NS      a.root-servers.net.
.                    IN  NS      b.root-servers.net.
a.root-servers.net.  IN  A       10.0.0.10
a.root-servers.net.  IN  AAAA    fd00::10
b.root-servers.net.  IN  A       10.0.0.11
b.root-servers.net.  IN  AAAA    fd00::11
arpa.                IN  NS      a.root-servers.net.
arpa.                IN  NS      b.root-servers.net.
cz.                  IN  NS      a.ns.nic.cz.
a.ns.nic.cz.         IN  A       10.0.0.20
a.ns.nic.cz.         IN  AAAA    fd00::20
sk.                  IN  NS      a.tld.sk.
a.tld.sk.            IN  A       10.0.0.25
a.tld.sk.            IN  AAAA    fd00::25
```

D OBSAH PRILOŽENÉHO DVD

- Elektronická verzia práce (PDF),
- Zdrojové súbory textu práce (LaTeX) upravované v prostredí ShareLaTeX.com.